RasPi

DESIGN
BUILD
CODE

15

Get hands-on with your Raspberry Pi

ASTRO PI

Track a space station

Welcome

Did you know that a Raspberry Pi is blasting off to space in just a few weeks? Part of the Astro Pi competition, it'll be sent up to the International Space Station with astronaut Tim Peake and a new add-on called the Sense HAT, plus code that has been written by primary and secondary school kids. We spoke to the teacher of one of the winning code clubs, Dan Aldred, to find out more about the competition and Thirsk School's winning entry – an ISS tracker – and we've even got a tutorial for you on how to re-create the project. Also in **RasPi** this month, we've got a great guide to adding touch and gesture control to your Pi with a brilliant board called Hover, so you can interact just by waving your hand!

Gavin Thomas

Editor

Get inspired

Discover the RasPi community's best projects

Expert advice

Got a question? Get in touch and we'll give you a hand

Easy-to-follow guides

Learn to make and code gadgets with Raspberry Pi

From the makers of

LinuxUser & Developer

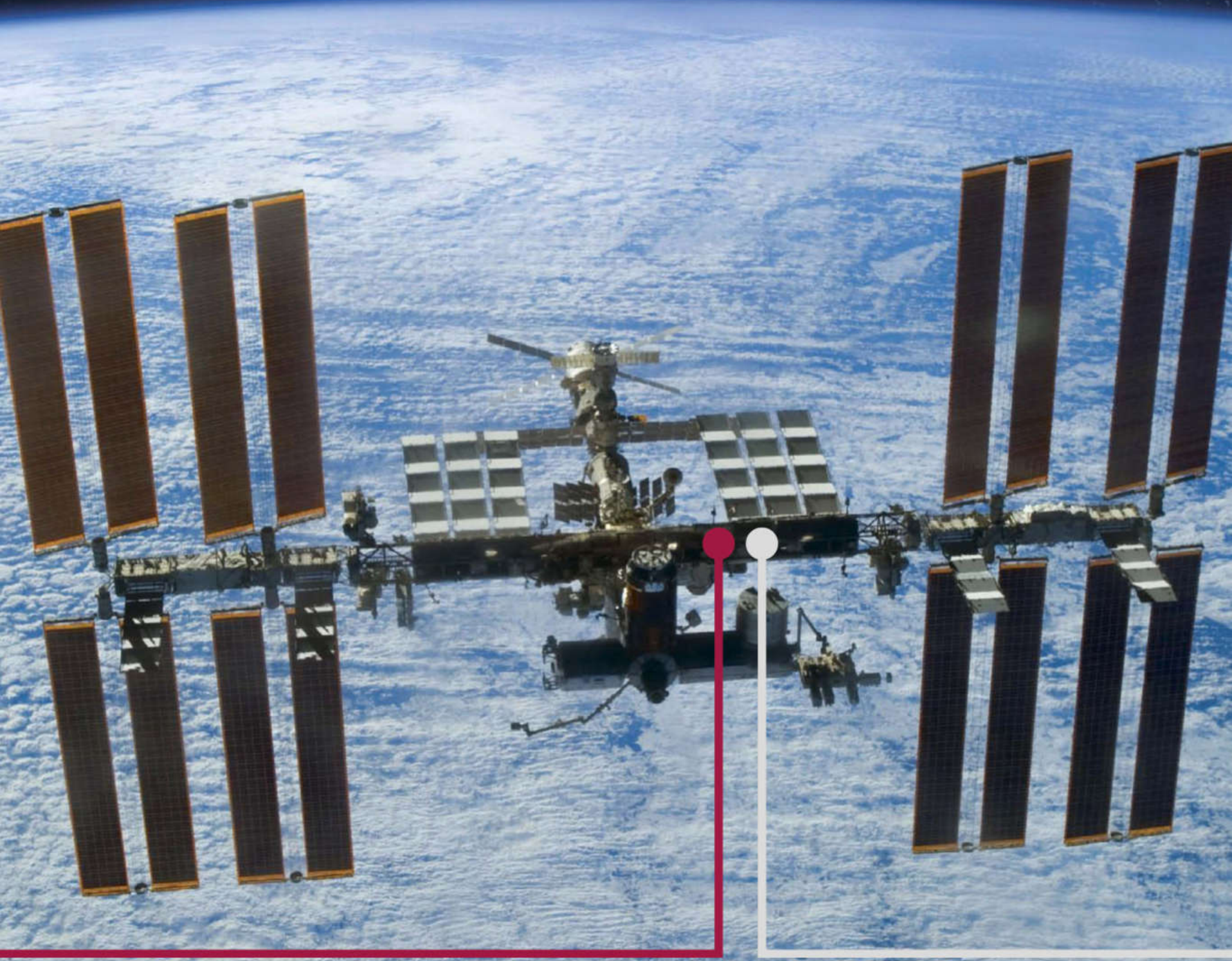
Join the conversation at...

@linuxusermag | Linux User & Developer | RasPi@imagine-publishing.co.uk

Contents

Astro Pi: Sending code to space

Clever Year 7 students at Thirsk School have made an amazing tracking system for the International Space Station. We speak to their teacher to find out more...



Can you tell us more about your students at Thirsk School who won the competition?

It was actually a code club that I'd set up at lunchtimes. The original reason for setting it up was to give students, who were perhaps what we call vulnerable learners, something to do at lunchtime – students who would struggle being in the playground; maybe their behaviour means they would get into difficulty, or they were just a bit more timid and so didn't have anywhere to go. Also, I was keen on making sure that the coding and the Raspberry Pi wasn't about bright kids – I wanted to make sure that low-ability kids and special needs kids had access to coding and all the benefits that it offers.

So I set up a coding club for lunchtimes, started with *Minecraft*, Sonic Pi, picamera photo hacking, and then this competition came along and I said, "Look, this is the opportunity we've got: a space rocket's going to go up to the ISS with an astronaut and an Astro Pi. What do you think?" They were like, "Yeah! Let's do it, let's do it!" And it grew from there – we ended up with eight to ten students who stayed every lunchtime for seven weeks, creating their winning solution.

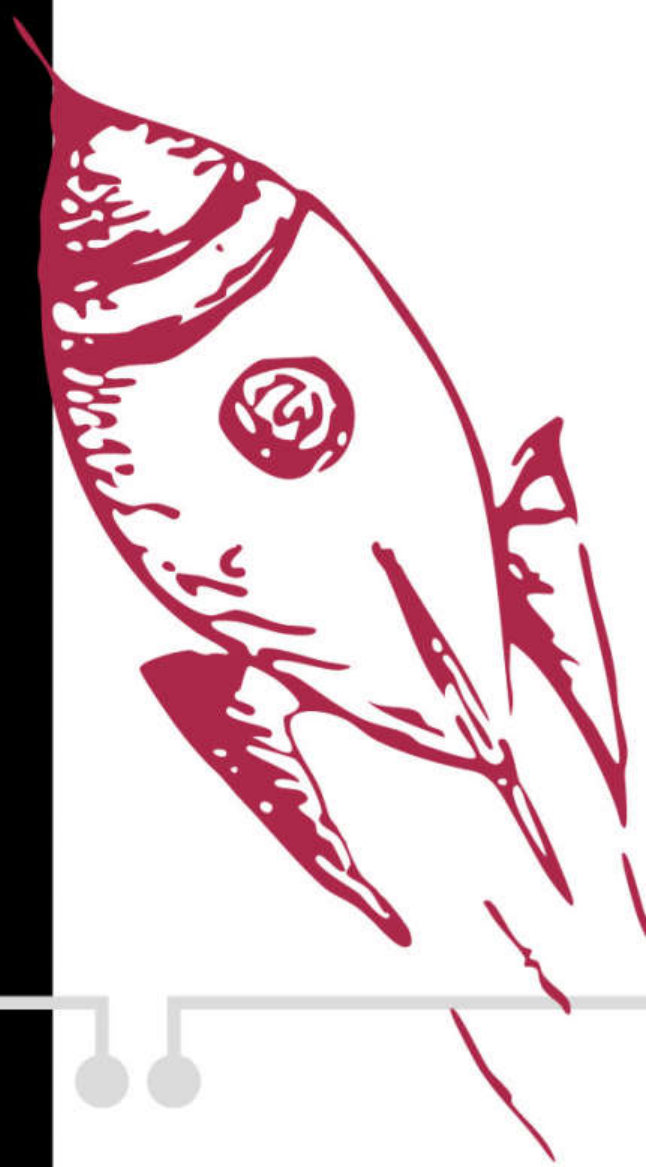
That's amazing dedication!

It is! In the end it became quite social, and by about week four they could see the results of what they'd made and start to get excited, thinking that it could actually win. But yeah, the dedication from them was huge, really motivated.

It must have been great for building a sense of community too, particularly with the vulnerable learners.

It was very exciting and rewarding personally, too. We started off with a shared document, so all the students could access the code from home, and what I found was

Dan Aldred is a Raspberry Pi Certified Educator and a Lead School teacher for CAS. He led a winning team of the Astro Pi Secondary School Contest and his students' code is being sent to the ISS in December.



Left Here are some of the Space-Byrds, the team behind the awesome ISS tracker

that as the weeks went on, the students were logging in more frequently to check their code, add their sections, and then they started editing each other's code. It was so polite – they'd come in at lunchtimes, for example, saying, "I noticed an error in your code last night. I changed it – hope you don't mind?" And then of course they had a common goal they could talk about, and they started talking about space and physics, different space films they'd seen, and of course as we were creating it they were talking about the different countries, whether they'd been to that country, what it's like, what's the capital – at work we call them learning conversations; they were learning just through talking around the subject.

Organic, peer-to-peer exchange.

Exactly – it wasn't manufactured. It was completely natural, which was absolutely brilliant. But yeah, they've forged some quite good friendships. And confidence as well – these are students who perhaps at the beginning when they started school (they were Year 7 students, so they're 11 years old now) wouldn't really go into the

"They had a common goal they could talk about, and they started talking about space and physics, different space films they'd seen, the different countries"

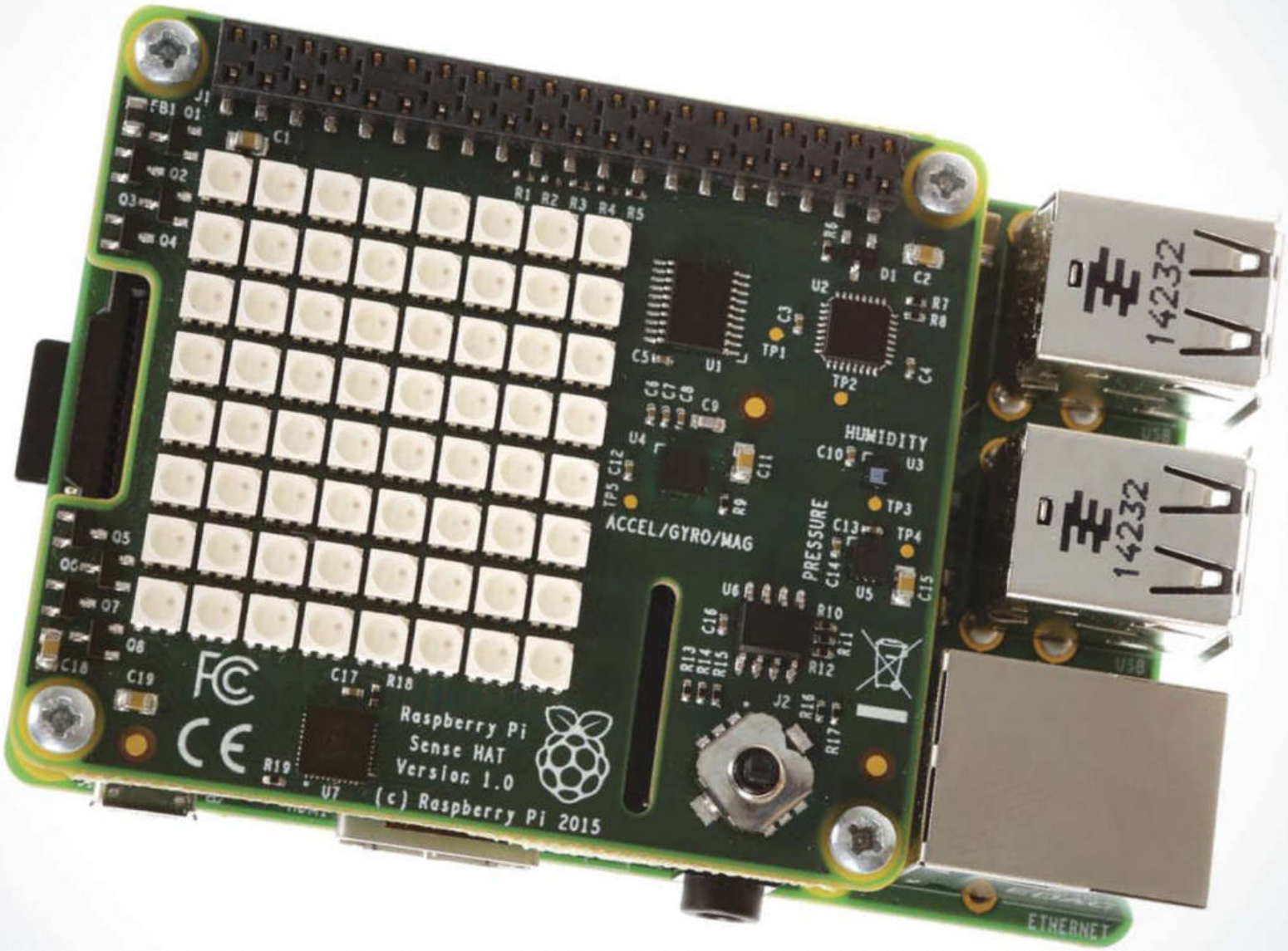
playground, were perhaps fearful of the dining hall, were maybe bottom-set students struggling with maths and English, or had a behaviour issue, and suddenly they've got quite a good status now amongst Year 7. And obviously the press have gotten hold of it and various local papers have run pieces on it and printed pictures of them, and I think it's given them a real boost. Rather than being labelled as an underachiever, a pupil premium child, having the potential to achieve, etc – well, they have all actually overachieved now!

It must have been amazing for their confidence and social skills, working in a collaborative environment like that.

Definitely. The program that they made was actually very simple in essence – it's just the fact that it's so big that took the time. In terms of the coding, it wasn't particularly difficult; it was just a case of there being 96 countries filled with 96 different flags, and 96 different languages that you have to find out and create messages for. So once they'd mastered the skill they were learning, it was then a case of repetition and discovery. The bit that was individual at that point was that the flag for Kazakhstan is different to the flag for the UK, and stuff like that. But creating each flag is a similar set of code – obviously the colours are slightly different, and the setup, but in essence the code was the same, so they could support each other, and say, "Oh, actually you've missed this bit out on the red, green and blue – you haven't got a value for the blue, that's why it's this colour." So yeah, they've learned a heck of a lot of skills and they've also learned a lot about the other countries as well, through supporting each other.

What is the Astro Pi competition?

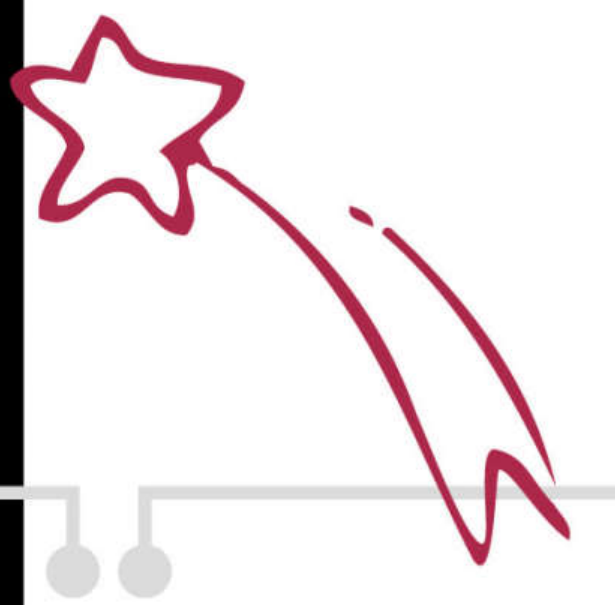
The Raspberry Pi Foundation and leading UKspace companies set primary and secondary school students the challenge of creating an innovative project using the Raspberry Pi and a specially designed Sense HAT module, which is packed with sensors and a colourful LED display matrix. The winning code will be sent to the ISS in December this year.



In terms of the logistics, how did the division of the work happen at the beginning and the end of the project?

There were two parts to the competition: the first was to pitch an idea, and you were then selected from that to go into the second stage. So the first couple of lunchtimes it was basically just brainstorming ideas, listening to what everybody wanted to come up with. We had some fantastic concepts, like, "Can we strap it to the astronaut, so that when he or she goes outside the ISS it can check for radiation?" Despite having the great ideas, we didn't quite know how much of it was realistic! I contacted Raspberry Pi and asked for a breakdown of what we can and can't do, and when we got the breakdown it said it was going to be stationary, it was going to be inside the station, it's not going to be moving, there's going to be no screen and the astronauts really need to have minimal

Above This is the Sense HAT – you can buy one for just £23: **http://bit.ly/1PWWZsU**



interaction with it, other than maybe starting it up and pressing a couple of buttons. So then we could shrink down the list, and I suppose the final idea came out because one student said, "So they're in space… how do they know where they are?" We talked about the different instruments and the fact they've got GPS or an equivalent tracking and co-ordinating system, but when they look over a country, how do they know which one they're looking over? And that's where the idea came out – why don't we have our Astro Pi system show the astronauts the flag of the country and a message, so they could compare that with the instruments on-board the space station and see if it works? So they all decided on that, we pitched it to Raspberry Pi, who said it was a great idea and sent us the kit, we got started, and picked out 96 major countries. For that, the students used the ISS trackers online and basically looked at the plot map of where it goes. It was quite a time-consuming process because they had to write down all the countries they were going to complete and put them into a shared Word document. I then put the example code at the top for England with the UK flag – from there they just had to work up the countries. Towards the end of the project we had a couple of students who'd set up a spreadsheet with all the 96 countries, 96 flags, 96 messages, and they began ticking them off.

And we had a couple of Astro Pis – one to test the flags and then the other was running all the co-ordinate tracking, so some of the students began working on that. It was probably by week five that we started to integrate the two together, so that if the ISS positional data was within the boundaries of the country then the flag pops up. Towards the end we could start to refine the longitude and latitude so that you got an exact position for the country.

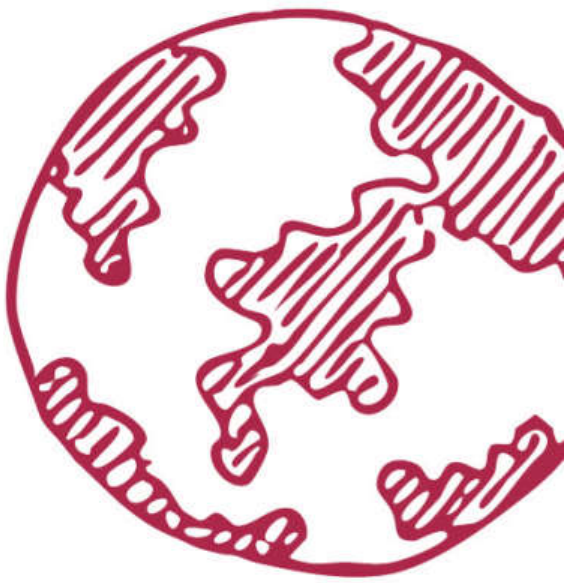
"That's where the idea came out – why don't we have our Astro Pi system show the astronauts the flag of the country and a message?"

One student was in charge of finding out all the longitudes and latitudes for the countries – an absolutely painstaking job because there were four points of origin for most countries, and there are some countries in L shapes so we had to do six or eight points. It's not perfect – it's quite a crude model and we're looking at a way of making it more accurate – but for the purpose of saying we're over Australia, for example, if you're within these four points of longitude and latitude then you're within the boundary. So one student was responsible for that.

So where exactly is the Raspberry Pi getting all of the longitude and latitude data from?

Here's the official press release of it: "the program uses telemetry data provided by NORAD along with the real-time clock on the Astro Pi to computationally predict the location of the ISS so it doesn't need to be online. It then works out which country's territory the ISS is above and shows its flag on the LED matrix along with a short phrase

"For the purpose of saying we're over Australia, for example, if you're within these four points of longitude and latitude then you're within the boundary"



Left This is Tim Peake, the British ESA astronaut who'll be taking the projects into orbit

in the local language". So that's the official blurb. The coding bit for the flags etc was tricky, but the mathematically challenging bit was the TLE file, which was a Two-Line Element file that looks at the time on the Raspberry Pi and makes a calculation of where the ISS should be. From that it returns the longitude and latitude position. The students wrote conditional statements – if it's within this longitude and latitude then it must be over this country, and therefore return this flag; and if it's not then it displays a little graphic and says 'calculating current position'. The experiment was comparing that set of results off the Raspberry Pi with what the ISS tracking system actually says on-board. It makes 19 orbits a day and can go out of sync, so the TLE file is updated 19 times a day. You have to download those two lines of code, pop it into your Python program and then it calculates the new positions. One of the biggest challenges was getting the time correct, but the Raspberry Pi Foundation has been great – it worked with us to ensure that it's accurate when the Raspberry Pi boots up, that the Astro Pi and Raspberry Pi are in sync, and that it's the correct time.

What's the next step for the project, then – are you completely ready for launch day, just waiting for Tim Peake to go up?

Yep – Raspberry Pi has been in contact. Tim's going up in December but on the 11th August he started doing a test run in Germany, which basically involves him being in a simulation for a number of weeks, and within that simulation he will run a number of experiments, including our ISS tracker experiment. So the code at the moment, the project we've built, is staying as it is and it's going to be used as a test run so Tim can check it works, that there's

"Tim's going up in December but on the 11th August he started doing a test run in Germany, which involves him being in a simulation for a number of weeks"

Above This is the flight case that will protect the hardware from excessive heat

no malfunctions, etc. And then in December he will fly up to the ISS and begin experiments there for six months, sending the data back to the UK.

So at that point, will you be running the experiment concurrently with an Astro Pi at Thirsk School?

Yep – as soon as we get confirmation he's on board, we're going to set up a copy of the ISS tracker and record the data from each day, and then with the two pieces of data that he returns – the ISS' real position from their flight instruments and then our ISS Astro Pi tracker – we'll compare all three.

In terms of September when we return to school, the maths department are on board now and they are going to build us a pixelated map of the world, where each element of the country boundary will be within a pixel grid reference, so what we can actually do is take

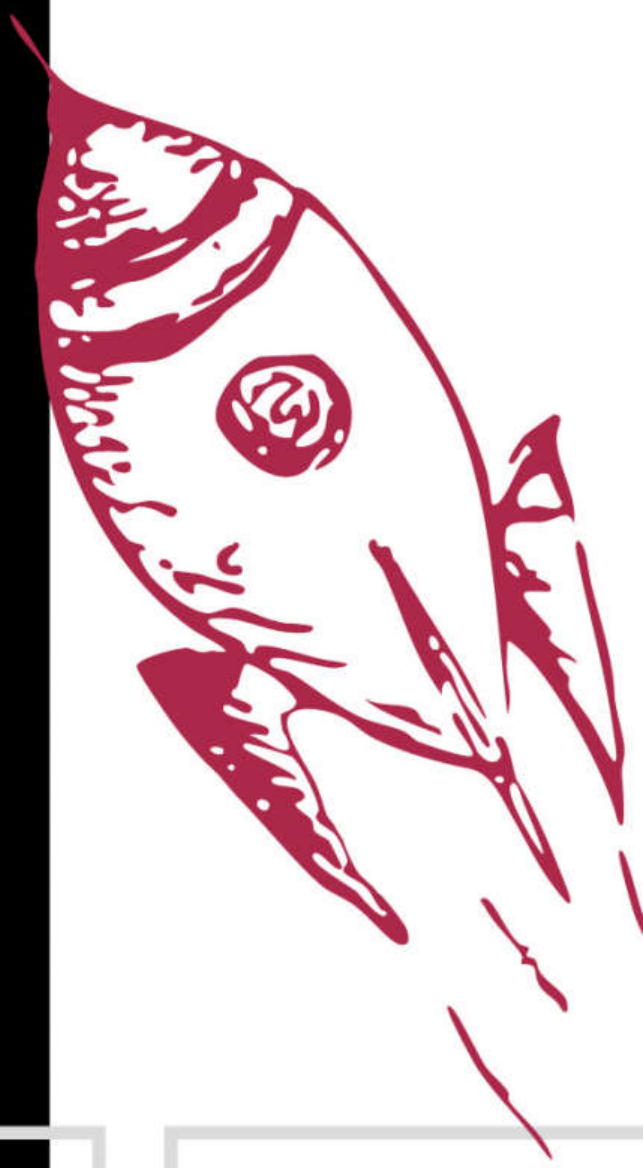
the longitude and latitude of each country and break it down to a pixel position. At the moment, what we've had to do for ease of use for the students is basically draw rectangles or squares around the countries using four points of origin, or with countries like Columbia, which is L-shaped, we've drawn a rectangle at the top and a rectangle at the bottom to get six points. So it's accurate, but with somewhere like Russia and Kazakhstan, as it goes over it actually undulates between the two different countries, so for two minutes it's in Kazakhstan and then for two minutes it goes into Russia and back out again. So for that kind of thing, our measurements weren't accurate enough to show that, but obviously a pixelated version of the atlas is going to be better.

I bet you'll have an awesome live-updating map going once you've got the pixel map sorted!

That's a good idea… I'd also like to set up some kind of live web feed so that everyone can compare the live ISS data with what our live Astro Pi ISS tracker is saying. A lot of the parents have contacted me, saying, "This is great – my son/daughter is talking about this and they're so excited." I'm going to share some pictures on Facebook and Twitter because I think when people actually see it, they'll understand it better. If I put a picture of some LEDs showing the Brazilian flag and say it's tracking the ISS, it doesn't really mean a lot. But if you can see there's the ISS over Brazil, and here's the Astro Pi with the Brazil flag, and now it's going over Columbia you can see the flag change, and oh there's the language…

When it started, the club was just running every Monday – now we're up to every lunchtime, five days a week. And we've got a beginner's club on Monday, so

"I'd also like to set up some kind of live web feed so that everyone can compare the live ISS data with what our live Astro Pi ISS tracker is saying"



what happens is the students who've been doing it since November last year come along and they support the new kids, and they feel really good now because they know everything – sudo idle and all the different commands – and they remember how they were when they first started. And they don't go to the club saying, "I'm going to learn coding." They go there saying, "I want to build a car that we can control from the computer. I'm going to build a tank. I'm going to play the *Mario* theme tune in Sonic Pi. I'm going to turn the water to ice in *Minecraft* just by walking on it." And that's what inspires them to do it. Exciting, isn't it?

Want to keep reading about this fantastic project? We couldn't fit the whole conversation into this article but you can read the uncut version of this interview online: **www.linuxuser.co.uk/news/astro-pi-space-byrds.**

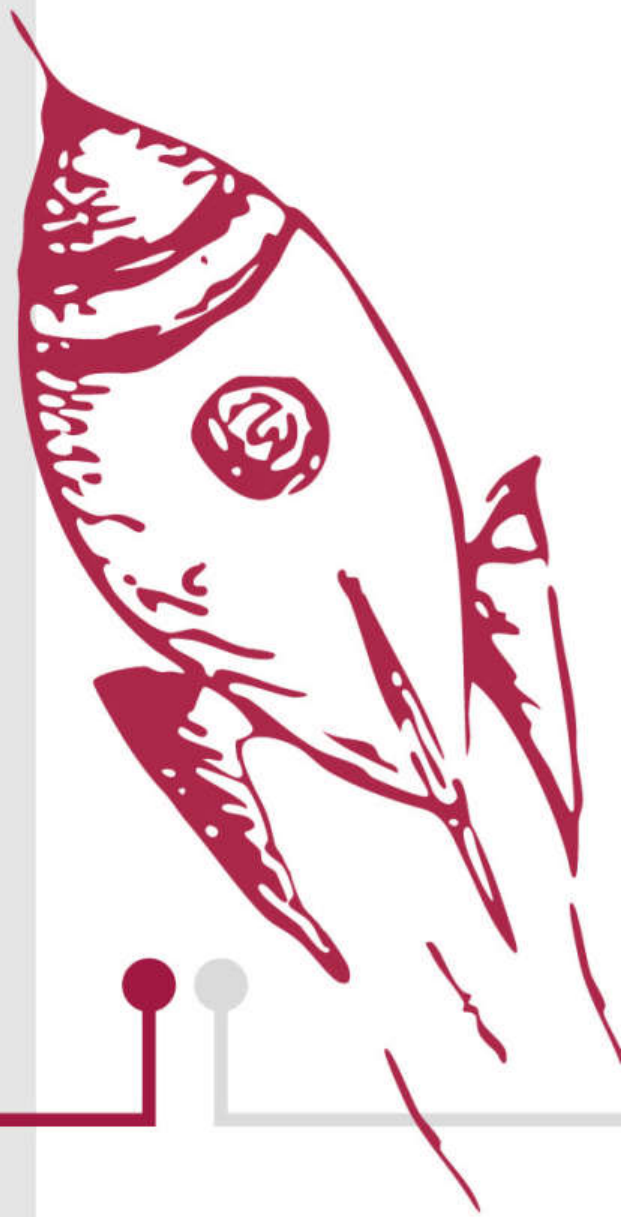
Below The LEDs in the matrix can be individually colour-controlled, enabling some cool graphics

Track the ISS

Use your Raspberry Pi and a Sense HAT to program the Astro Pi-winning ISS tracker

As part of his six-month mission, Tim Peake will run a number of Astro Pi experiments created by schools, each judged and selected by the ESA, the National Space Agency and Raspberry Pi.

The Sense HAT was designed and built specifically for this mission, and boasts an array of sensors and an 8x8 LED matrix. Each experiment will generate and collect data which will then be downloaded to Earth for analysis. In this tutorial, you will be introduced to some Astro Pi programs and learn how to create a program to track the longitude and latitude of the ISS in real time. If you do not have an Sense HAT, skip ahead to Step 9.



01 Install the Astro Pi software

Attach the board to the GPIO pins and install the Astro Pi software, downloadable from the hosted Astro Pi website. Boot up your Raspberry Pi, load the LXTerminal and type in the following code on a single line. On completion, reboot your Raspberry Pi by typing sudo halt:

```
wget -O - http://www.raspberrypi.org/files/astro-pi/astro-pi-install.sh --no-check-certificate | bash
```

02 Example programs

The software comes with a few starter programs that can be used to test that the Astro Pi is functioning correctly and to demonstrate some features of the board. The example programs are stored inside the /home/pi/astro-pi-hat/examples folder and run in Python 3 IDLE.

"The software comes with a few starter programs that can be used to test that the Astro Pi is functioning correctly and to demonstrate some features"

03 Take a temperature reading

The Astro Pi has a built-in thermometer that can be easily coded to read and return the current temperature. The sensor is fairly close to the CPU and may pick up some residual heat, however on the whole the reading is sound. To measure the temperature on the Astro Pi, open your Python 3 editor and type in the code below, then save and run it. The current temperature reading will be returned in the shell.

```
from astro_pi import AstroPi
ap = AstroPi()
temp = ap.get_temperature()
print("Temperature: %s C" % temp)
```

04 Compass reading

One of the nifty sensors on-board is the compass. This can be used to return a measurement of the Astro Pi's position in relation to magnetic north. The code is simple to use: ap.get_compass() in line 3 (next page) returns the position which is stored in a variable called north. The value that is measured is then printed out in line 4. Use this code example to test the compass sensor and the readings:

```
from astro_pi import AstroPi
ap = AstroPi()
north = ap.get_compass()
print("North: %s" % north)
```

05 LEDs

The 8x8 LED matrix is programmable and includes a range of colours and brightness settings. Each LED can be coded individually and combined to create a simple image. To set an LED colour, create a variable and assign an RGB value to it. In line 3 (below) the colour is set to red, using the values (255, 0, 0). Add additional colours by creating additional variables and setting the RGB codes for each new colour. Then create a representation of the image using the variable names – in this example, the X and O symbols (line 6) combine to create a question mark. Set the LEDs with the code ap.set_pixels(question_mark) in line 7.

> "Each LED can be coded individually and combined to create a simple image. To set an LED colour, create a variable and assign an RGB value to it"

```
from astro_pi import AstroPi
ap = AstroPi()
X = [255, 0, 0]  # Red
O = [255, 255, 255]  # White
question_mark = [
O, O, O, X, X, O, O, O,
O, O, X, O, O, X, O, O,
O, O, O, O, O, X, O, O,
O, O, O, O, X, O, O, O,
O, O, O, X, O, O, O, O,
O, O, O, X, O, O, O, O,
O, O, O, O, O, O, O, O,
O, O, O, X, O, O, O, O
]
ap.set_pixels(question_mark)
```

Above Converting your own images is a great way to speed up the creation of LED matrix graphics

06 LED per pixel

The image on the LED matrix can also be set automatically from an image file. For example, an image of a space invader can be loaded, the colours and positions calculated and then the corresponding LEDs enabled. Ensure that your image is 8x8 pixels in size and save it into the same folder that the program is saved within. Use the code below to open and load the image of the space invader – the Astro Pi will do the rest of the work:

```
from astro_pi import AstroPi
ap = AstroPi()
ap.load_image("space_invader.png")
```

07 A single letter

The LED matrix can be used to display a single letter using the simple code line ap.show_letter(str(a)) – this code would display the lowercase letter 'a' on the matrix. Using a for loop and a range function (line

4), you can create a simple countdown that displays numbers from 9 to 0. Note that the list is reversed; this enables the numbers to count down from 9 to 0.

```
import time
from astro_pi import AstroPi
ap = AstroPi()
for i in reversed(range(0,10)):
    ap.show_letter(str(i))
    time.sleep(1)
```

08 Scroll a message

Writing code to scroll text on LCD/LED displays can be challenging and frustrating. The Astro Pi API removes the difficulties and simplifies the whole procedure to a simple line of code: ap.show_message("This is a test message"). Change the text between the quotation marks, save and run the program, and your message will then be scrolled across the Astro Pi LEDs. Adjust the colour of the message and the time it takes to scroll by including text_colour=[255, 0, 0], setting an RGB value, and scroll_speed=(0.05) within the function's brackets. Try experimenting with this example code:

```
from astro_pi import AstroPi
ap = AstroPi()
ap.show_message("Linux User and Developer",
text_colour=[255, 0, 0])
```

09 Install PyEphem

The remaining steps cover the program to track the ISS in real time. PyEphem provides astronomical computations for the Python programming language. Given a date and location on the Earth's surface, it can compute the positions of satellites whose orbital elements the user

Pixel perfect

The Astro Pi's LED matrix is 8x8 in size and there are several websites and apps that can be used to mock up an image, modify and create a suitably sized image, for example: **gurgleapps.com/tools/matrix** or **piq.codeus.net/draw**

provides. The ISS is technically a satellite as it orbits the Earth, therefore the PyEphem library can be used to track it. Install the software using Pip:

```
sudo apt-get update
sudo apt-get upgrade
pip install pyephem
```

10 Import the required modules

For this and the following steps, refer to the annotations in the full code listing at the end of this tutorial. Open a new window in IDLE 3 and import the modules shown. These import the Astro Pi API, the position tracking program and the time functions to allow you to add pauses or rests to your program.

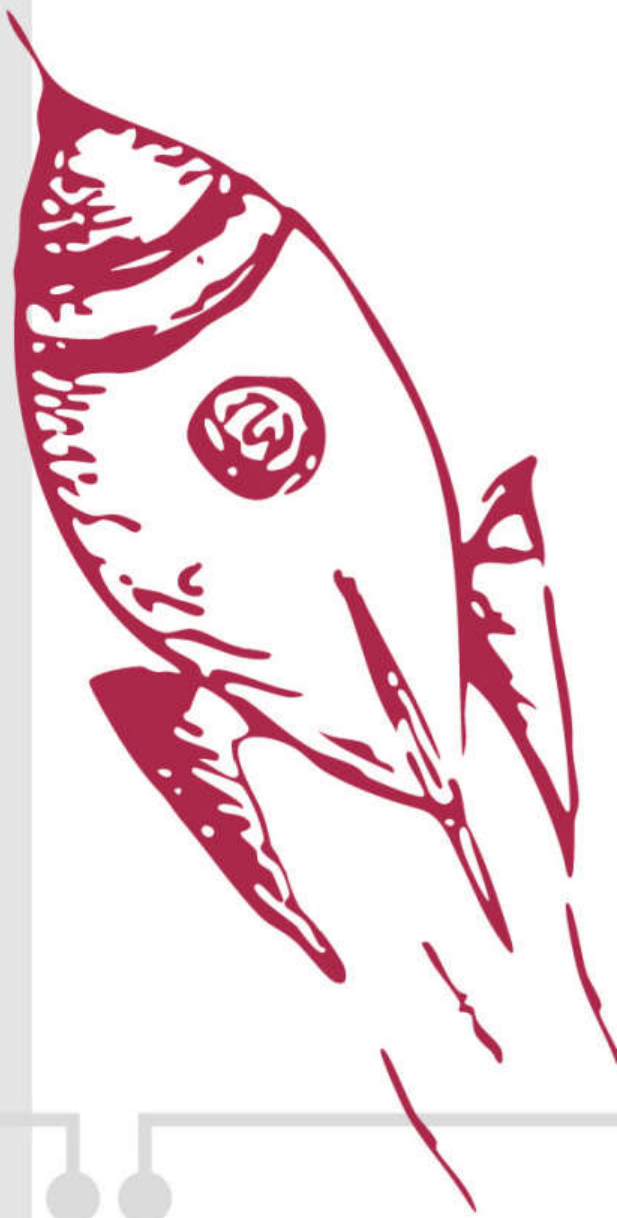
11 The TLE file

To calculate the position of the ISS you will need to use an up-to-date Two Line Element (TLE) file. The TLE is a standard mathematical model to describe a satellite's orbit and is processed by tracking software. The data results returned include predictions for viewing times, speed and the current position, returned as longitude and latitude values. The TLE data is available from the NORAD website and is updated several times a day: **https://celestrak.com/NORAD/elements/stations.txt**. Go to the site and copy the first three lines of data at the top of the page.

12 Prepare the TLE data

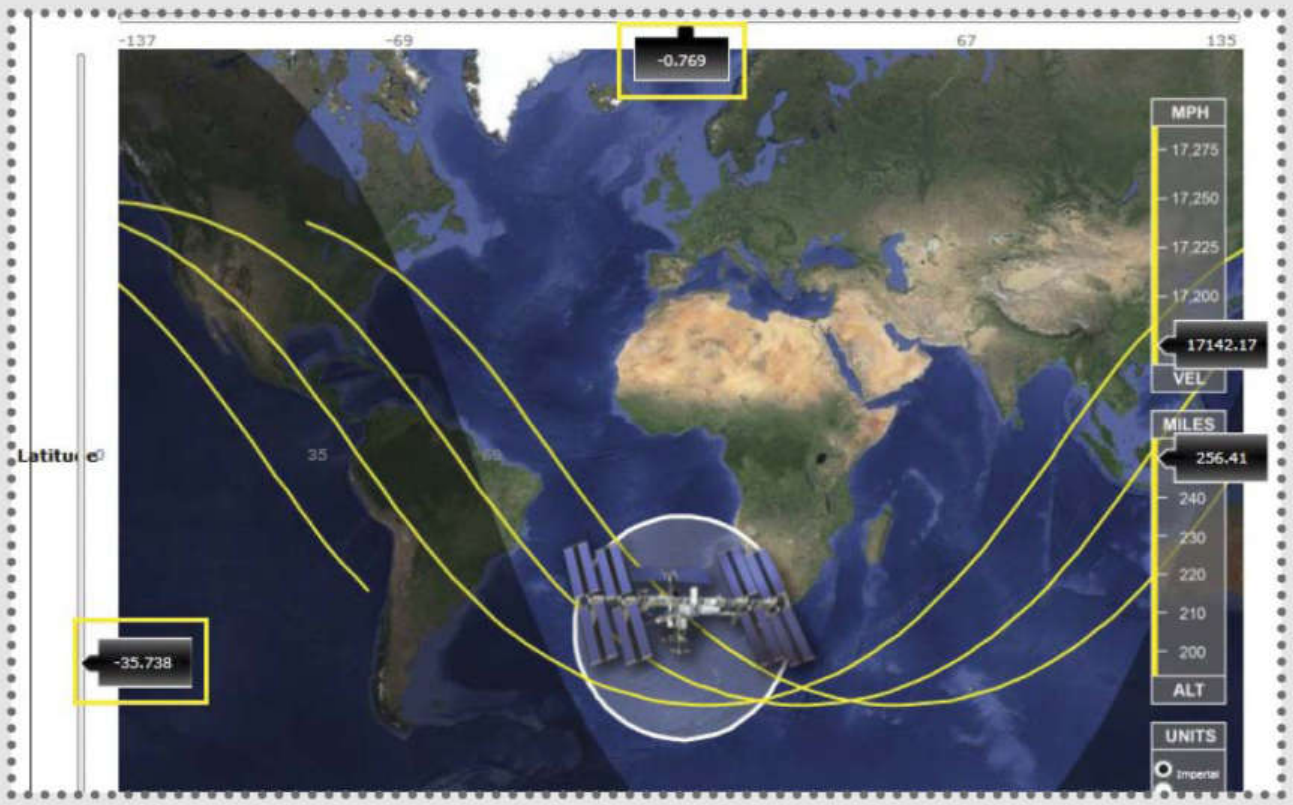
Before you can use the TLE data, you need to ensure that it is set up correctly – if it isn't then you will receive errors. In your Python program, create three new variables called name, line1 and line2. Next to the name variable

"The ISS is technically a satellite as it orbits the Earth, therefore the PyEphem library can be used to track it"



add the name of the satellite: ISS (ZARYA). Now add the data from line one of the TLE to the variable called line1. Do the same for line 2; adding the second line of data. Ensure that the layout of the variables remains the same, as shown in the full code listing.

The data shown in the full code listing is just an example; the current data values and their formatting can be found over at the NORAD site that we linked to in the previous step.



13 Calculate the position of the ISS

The TLE data is now ready to calculate and predict the position of the ISS. A further three lines of code will enable you to retrieve the data. The first line, tle_rec = ephem.readtle(name, line1, line2), creates a variable of TLE data. In line two, tle_rec.compute(), the maths is crunched and the position calculation is performed. Once this is completed, extract the required longitude and latitude measurement data using the line print tle_rec.sublong, tle_rec.sublat. You can compare the result with an online tracker like **www.isstracker.com**.

"The current data values and their formatting can be found over at the NORAD site"

Left Website-based ISS trackers also use the TLE file to plot the station's position

Remember that the accuracy of the TLE prediction is based on the clock time of your Raspberry Pi – ensure that this is accurately set.

14 Convert to a string and split

The data retuned is very accurate and you will note that the tle_rec.sublong, tle_rec.sublat data can be up to nine decimal places in length. You may find that this is too accurate for your measurements as most countries' longitude and latitude are given to a single decimal place. In order to reduce the decimal places, you need to convert the data to a string and split the data at the colon. Create two new variables and use str to convert the data to a string, as shown in the first two lines. Use split(":") to split and return usable data, as shown in the next two lines.

"Remember that the accuracy of the TLE prediction is based on the clock time of your Raspberry Pi – ensure that this is accurately set"

15 Print the data

Once the data is tidy and usable, convert it back into a float number. This is handy for using the values to check the location that the ISS is currently flying over and compare this with a country's boundaries (see Step 17). Convert the variables back into a float value using the code: lati[0] = float(lati[0]). In the first two lines, the [0] ensures that only the value is returned with the first decimal position. The next two lines check that the data returned is usable; they aren't needed in the final code.

16 County comparison

Now that you have the longitude and latitude positions for the ISS you can begin to compare these with the positions of cities, capitals and countries, plotting the location. There are many websites out in the wild that

Left There are plenty of ways to get precise latitude and longitude values of any location in the world

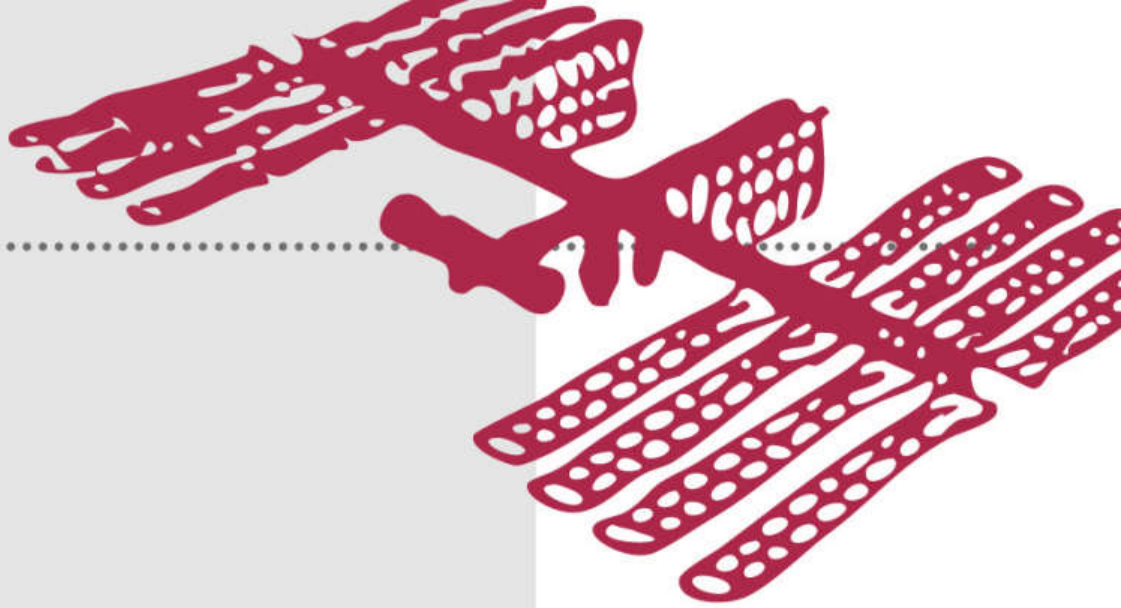
list the positions of a capital city – for example, try **www.csgnetwork.com/llinfotable.html**. You can use sites like **http://itouchmap.com/latlong.html** to plot the boundaries of a country in terms of its longitude and latitude. This is challenging, as some countries undulate between two or three borders. You will find it easier to take a rough approximation of the countries' shapes and co-ordinates.

17 Comparison with position data & country

The final step is to take the data and compare it with the country boundary data – ie if the ISS is within this range then it is within that particular country's boundary. Create a simple conditional using an if statement to check when the ISS flies over, say, the UK. Use a print statement to display the name of the country. You can also use the LED code from Step 5 to create a flag of the county that is displayed as the ISS flies over the country.

"Create a simple conditional using an if statement to check when the ISS flies over, say, the UK. Use a print statement to display the name of the country"

The Code ISS TRACKER

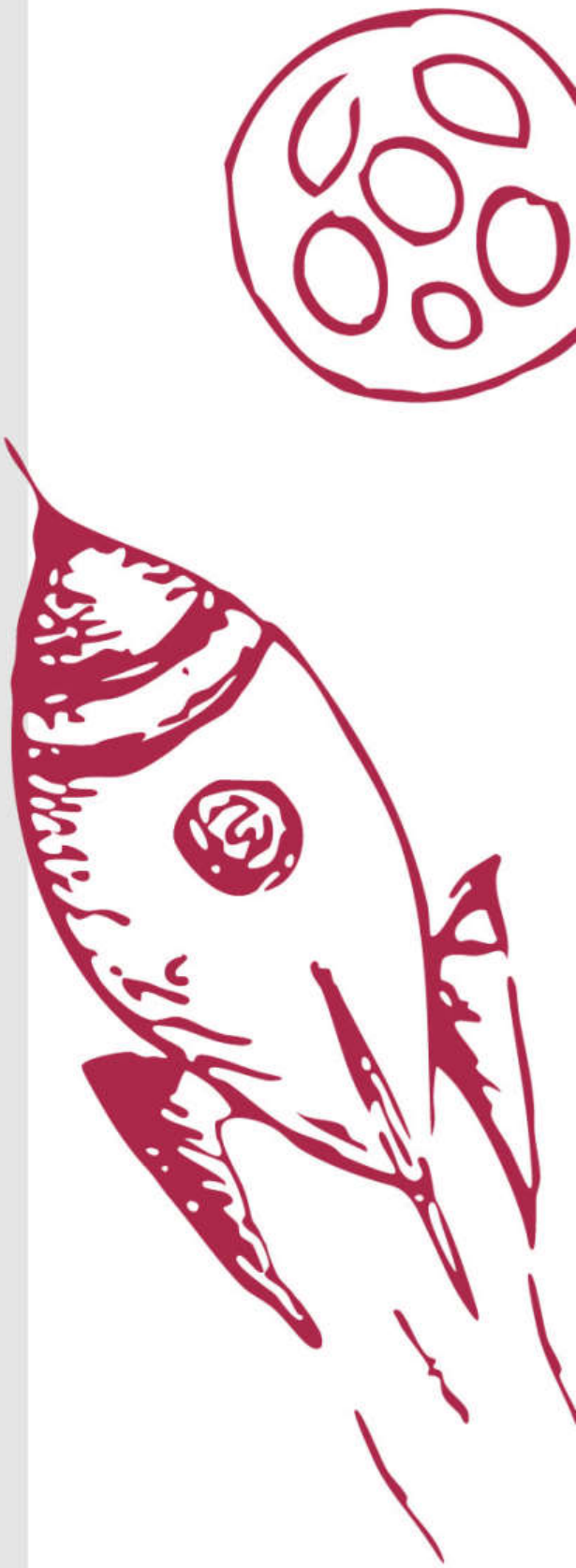


```
from astro_pi import AstroPi
import ephem
import datetime
import time
ap = AstroPi()

name = “ISS (ZARYA)”;
line1 = “1 25544U 98067A   15185.95963984  .00006354  00000-0  98170-4 0  9990”
line2 = “2 25544  51.6454 355.2696 0003202 121.3230  14.1346 15.55509232950800”

ap.clear()

while True:

    tle_rec = ephem.readtle(name, line1, line2)
    tle_rec.compute()

    #convert to strings#
    lat2string = str(tle_rec.sublat)
    long2string = str(tle_rec.sublong)

    #Split to pull out data
    lati = lat2string.split(“:”)
    longt = long2string.split(“:”)

    ###Convert to floats to check the ranges

    lati[0] = float(lati[0])
    longt[0] = float(longt[0])
    print lati[0]
    print longt[0]
```

The Code ISS TRACKER

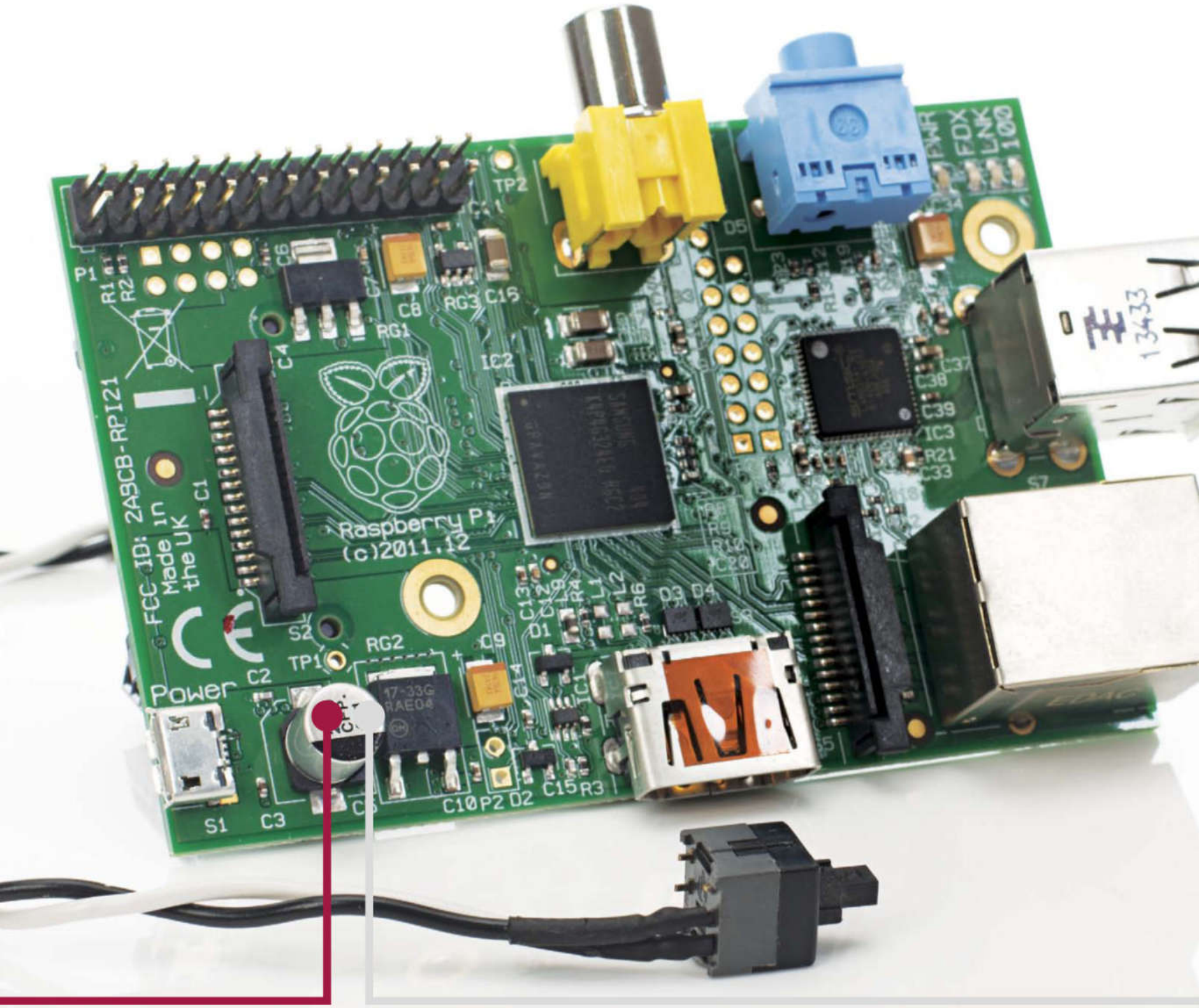
17

```
    ###Check the location###

    ###UK###
    if (lati[0] <= 53 and lati[0]>= 52) and (longt[0] >= -4 and longt[0]<= -1):
        print "United Kingdom"

        X = [255, 0, 0]  # Red
        O = [255, 255, 255]  # White

        UK = [
        O, O, O, X, X, O, O, O,
        O, O, O, X, X, O, O, O,
        O, O, O, X, X, O, O, O,
        X, X, X, X, X, X, X, X,
        X, X, X, X, X, X, X, X,
        O, O, O, X, X, O, O, O,
        O, O, O, X, X, O, O, O,
        O, O, O, X, X, O, O, O
        ]

        ap.set_pixels(UK)
        time.sleep(6)
        ap.show_message("Hello ISS, you are over the UK")

    else:
        ap.show_message("Checking location")
```



Install a reset switch on your Raspberry Pi

Need to restart your Pi after a system lock-up? Ease strain on the mains connector – install a reset switch!



Shutting down a Raspberry Pi by removing the power cable is risky. Data may be writing to the SD card, leading to corruption, while repeated removal of the power cable can cause problems with the connector port. Such faults cause the Pi to hang, so the simple fix here is to add a simple reset function to the device. There are three ways this can be done: with a USB reset button, a motherboard jumper on the GPIO bus or with a momentary button connected to newly-soldered pins on the P6 header on the Model B Rev 2 and B+ (this is the most complicated option).

If you have an old PC around, retrieving the reset button and cable from this (and even the connecting motherboard pins) is achievable if you're handy with a soldering iron. Otherwise, we recommend purchasing the parts online since they're cheap enough.

THE PROJECT ESSENTIALS

Momentary switch

Soldering iron and solder

Single pin pair header

HDD/motherboard jumper

01 Check your Raspberry Pi model

Only two models feature the P6 header: the Raspberry Pi Model B Rev 2 (which you can find next to the HDMI port) and the B+ (to the left of the '© Raspberry Pi 2014' label). You will need to install the pins manually, however, as they are not preinstalled for this function.

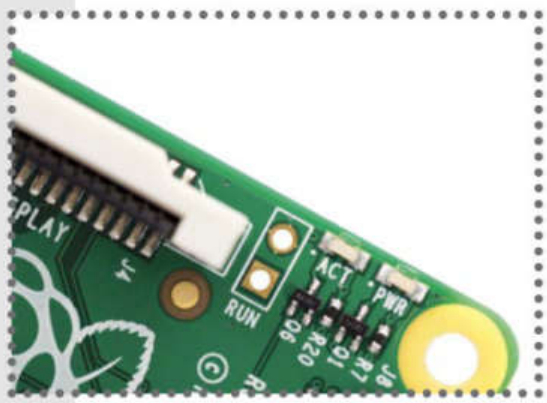
02 Find your components

Header pins can be purchased online, although this will invariably result in having to order more than you need.

Alternatively, if you have an old motherboard, remove a pair of pins with a soldering iron. Similarly, you might buy a new reset button, or simply use one from an old PC.

Above On the B Rev 2, look for two small holes near the HDMI

Below On the B+, the holes are beside the GPIOs, near the corner



03 Solder pins to your Pi

To gain stability when soldering, place the Pi upside down on a layer of packaging foam, with the header slotted into the holes.

Using fine solder, secure the pins to the mainboard with your soldering iron. This will require a very steady hand, so get assistance if required.

04 Connect your reset switch

Leave the solder to cool for a few minutes before attaching the reset switch connector. Some cases don't have space for the pins and/or the connector, however, so take the time to plan ahead and make sure everything fits. If not, you may need to make some adjustments to your case.

05 Reset Raspberry Pi following crashes

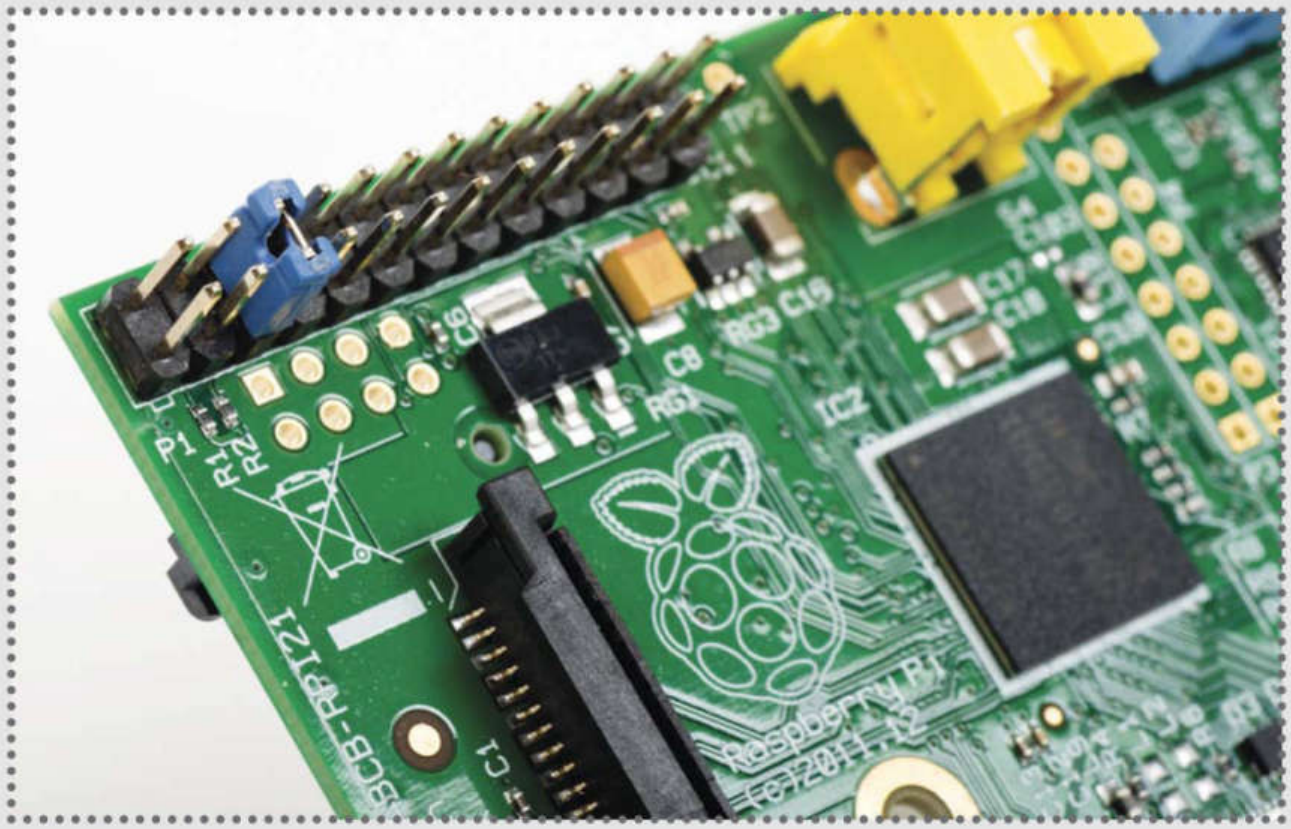
With the switch installed, you'll be able to reset the Raspberry Pi when required. Note, however, that this isn't an option to be used for whenever you feel like restarting. Rather, this method should only be used when the system fails to respond within a reasonable time frame.

06 Reset with a HDD jumper

Not keen on soldering new pins to your Raspberry Pi? That is perfectly understandable, but it doesn't mean you cannot reset the computer. We have another solution for you.

Using a motherboard jumper, two GPIO pins and a script to initiate an ordered shutdown is a simple alternative that doesn't involve any soldering and potential PCB damage.

"To gain stability when soldering, place the Pi upside down on a layer of packaging foam, with the header slotted into the holes"



Left If you're running low, Adafruit has a pack of 40 6-inch female/male extension jumpers for just four dollars

07 Identify the GPIO pins

This method works on most models. Each has a GPIO array, 26 pins on the A and B (Rev 2) and 40 on the A+, B+ and 2B. The jumper should be placed on GPIO3, pins 5 and 6, counting from the left with the board the right way around.

08 Detect jumper with a script

Run the script at **bit.ly/1Ge5n0O** to detect the jumper, first making it executable (**sudo chmod 755**). Within a minute your Pi will shut down. Add this line to /etc/crontab to run the script whenever you boot up:

```
@reboot    root    /home/user/scripts/gpio_actions.sh
```

Remember to remove the jumper before booting up!

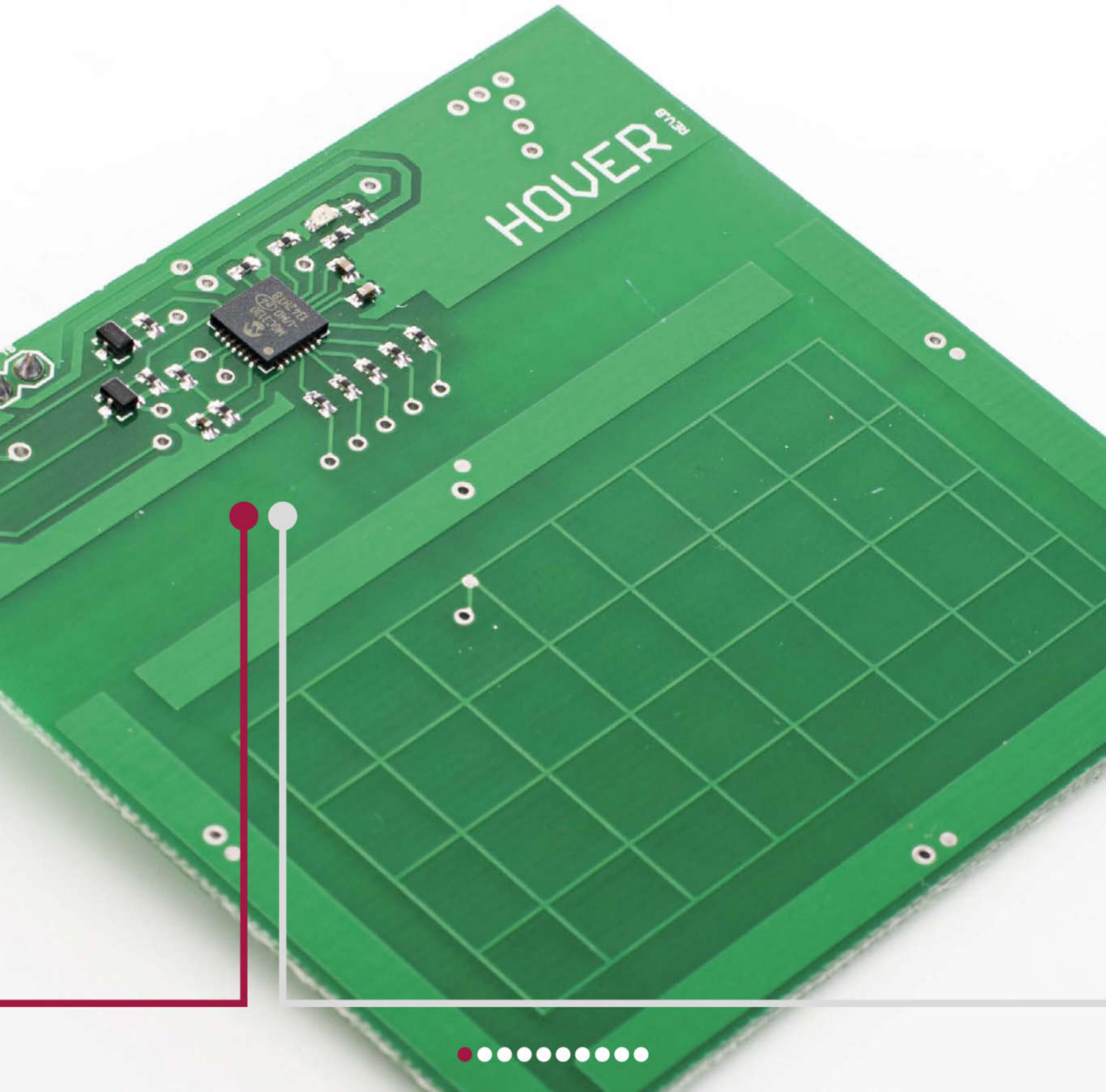
09 Try a USB reset button

Specialist online stores offer USB reset buttons that can be connected to your Pi for scenarios when the device needs to be rebooted. If the idea of using the HDD jumper or doing some minor soldering doesn't suit you, then a USB reset switch might be your best option.

"Specialist online stores offer USB reset buttons that can be connected to your Pi for scenarios when the device needs to be rebooted"

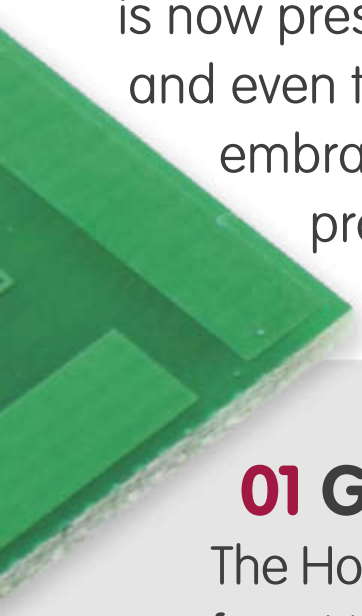
Add gesture control

Hover is an impressive add-on for your Raspberry Pi that allows you to add touch and gesture control



People often ask what the best way is for them to get started with Raspberry Pi. Obviously this does depend on the individual user and what they want to achieve and get out of any project, but in a more general sense it's often the hardware projects that win out for getting to grips with it. They teach a variety of skills (including programming, circuit building, soldering, hardware design and more) and are also varied enough to both keep beginners interested and allow them to work out for themselves exactly what aspect they love best. Even a seasoned professional will get a serious kick out of a bit of physical computing and automation! This is one of the unique features of the Pi compared to traditional "black box" computers; you can break out of the usual boundaries and interface with everyday objects like never before.

One of the most important aspects of a hardware project is often the user input mechanism, and as technology is refined we see new and more intuitive ways to accomplish this task. Gesture and touch control is now present in a large number of consumer devices and even the biggest technophobes are starting to embrace this technology. It is time to bring your Pi projects into the 21st century with Hover!



THE PROJECT ESSENTIALS

Hover board
bit.ly/1OgRMhK

Breadboard

Jumper cables

Speakers or headphones

> "Gesture and touch control is now present in a large number of devices and the biggest technophobes are starting to embrace this technology"

01 Get the gear!

The Hover add on board is available to purchase direct from Hover (**www.hoverlabs.co/#shop**) for $39 (£25), however this will ship from Northern America and therefore if you are based in the UK or Europe it will likely be quicker and cheaper to order from one of the other retailers listed via the above link. The added

benefit of ordering from a retailer is that if you need any other items you can likely get those at the same time!

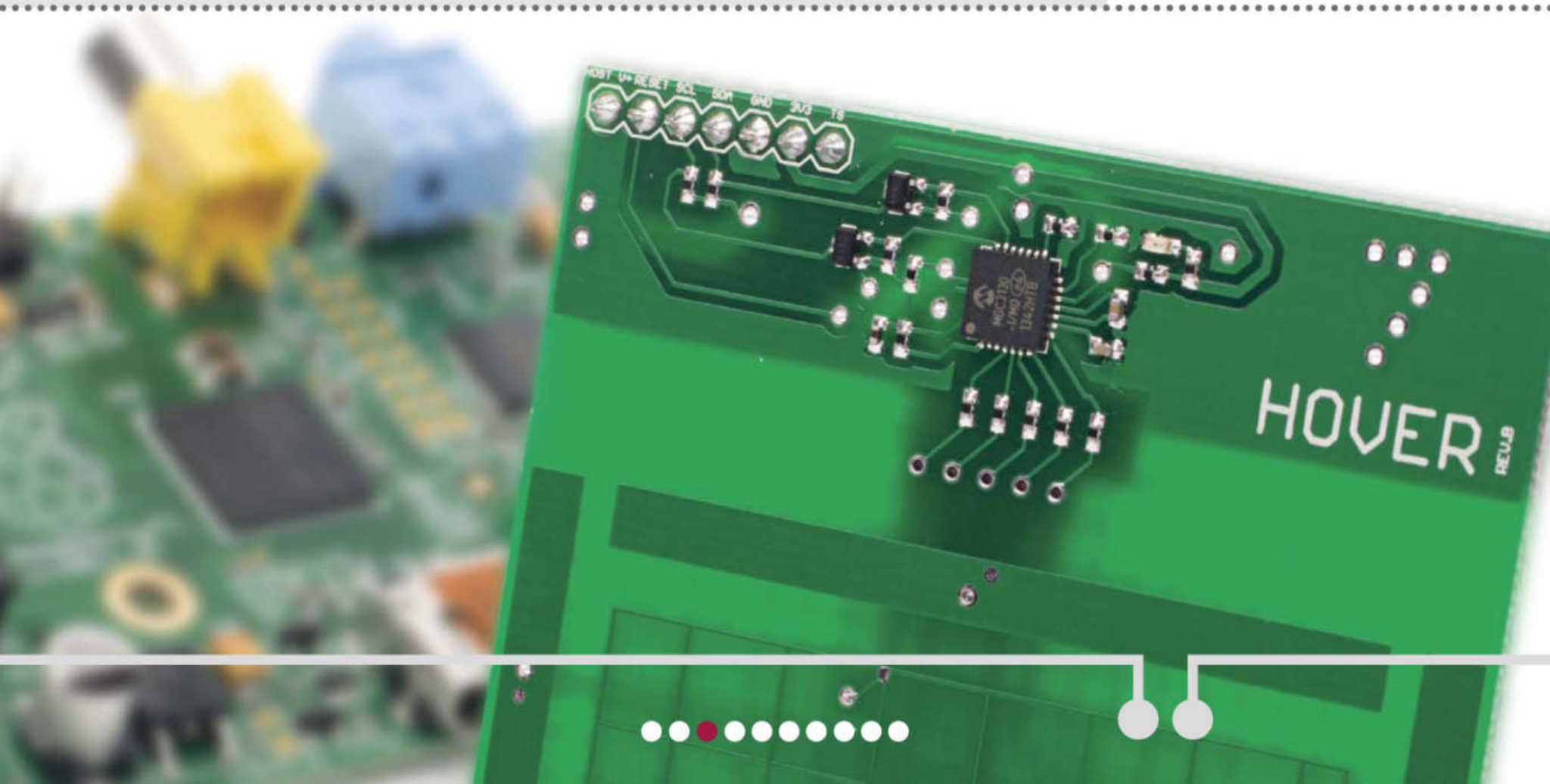
Hover will work perfectly with any Raspberry Pi, including both the new plus versions and the older models – just make sure your system is fully up to date with:

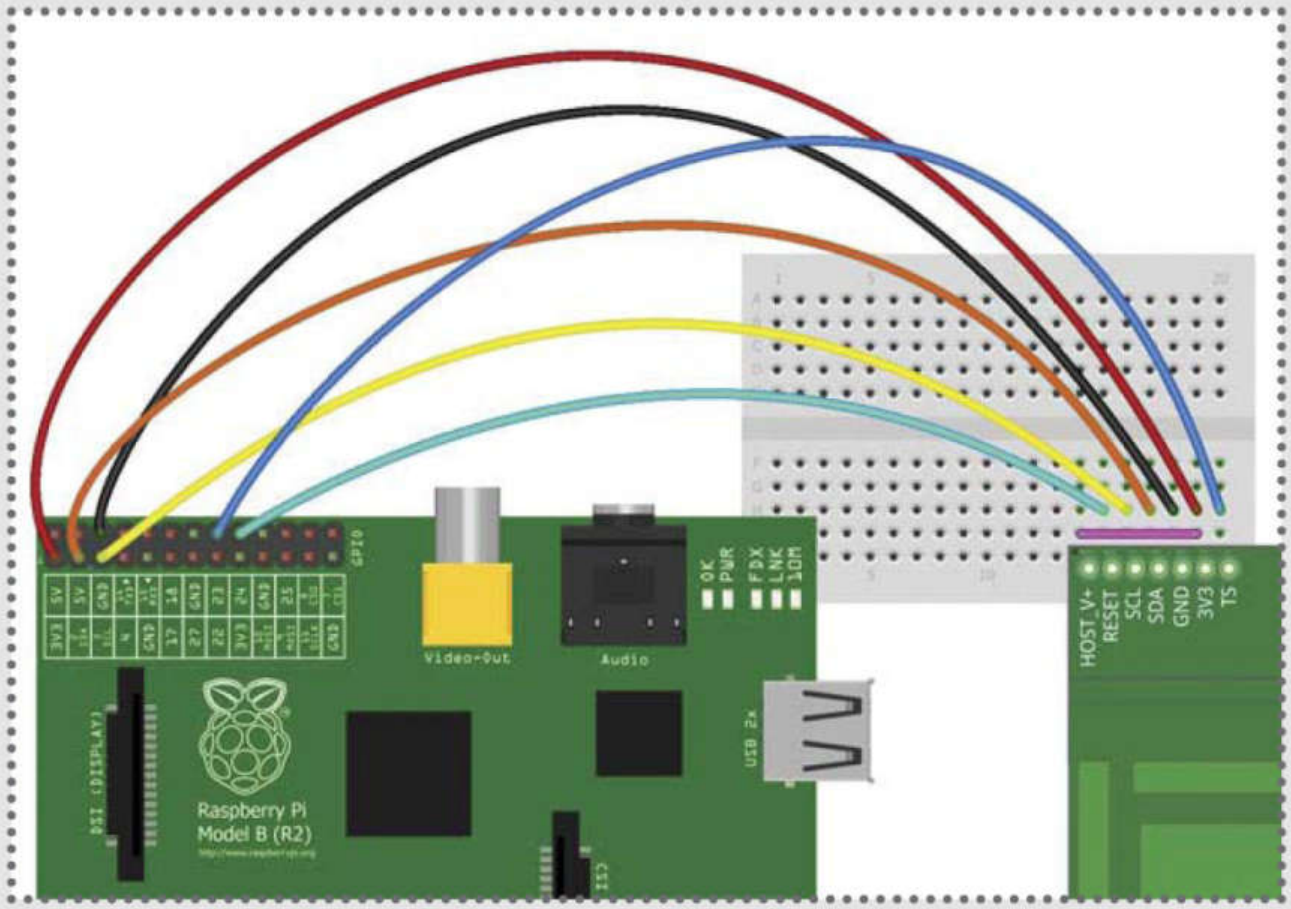
```
sudo apt-get update
sudo apt-get upgrade
```

02 Update GPIO and I2C

When making use of GPIO and I2C (or any other interfacing technique on the Raspberry Pi, for that matter) it is always good practice to first update to the very latest software versions possible. Newer software versions typically have bug fixes and additional futures which can come in very handy. GPIO and the RPi.GPIO Python library are installed by default on Raspbian, but you may need to enable I2C if you haven't done so already. This is a fairly standard process and has been covered many times, so we won't go into it here. We would, however, highly recommend the brilliant I2C setup tutorial from Adafruit (**bit.ly/1igFto4**).

Below Hover can recognise up, down, left and right swipes made in the air up to five inches away





Left There are quite a few wires to connect up to the Hover (shown on the right) but they're quite straightforward to do

03 Set up the hardware

Make sure your Raspberry Pi is powered down and not connected to power before starting this step, to avoid any unnecessary damage to your Raspberry Pi. Pick up your Hover, breadboard and wires then connect them as shown in the Fritzing diagram. The physical pins you should be using on the Raspberry Pi are 1, 3, 5, 6, 16 and 18. Whilst a Model B Pi is shown, this will be the same connection on a Model A, B, A+, B+ or 2B of any revision. Once completely set up like in the image, reconnect the power cord and open an LXTerminal session.

04 Check the connection

Hover connects to the Raspberry Pi through the I2C interface located on the main 26- or 40-pin GPIO bank (depending on which version of the Raspberry Pi you are using). There is a very easy way to check if your Raspberry Pi is correctly connected to Hover using the simple command line I2C tools. Issue the following command:

```
sudo i2cdetect -y 1
```

Plenty of platforms

The Hover board has intelligent on-board level shifting, meaning that it can be used with either 3.3V or 5V logic levels and therefore with pretty much any microcontroller. Connection examples and code snippets are available for Arduino, Sparkcore and PCduino on the Hover website, and these can also be adapted to suit other devices relatively easily.

If you see 42 in the response then you are successfully connected to Hover!

05 Using a Rev 1 Pi?

In the code, we have passed an option, **-y 1**, that tells the operating system which I2C bus to look at (there are two on the BCM2835 processor on the Pi). The first revision Raspberry Pi (the one that initially launched in February 2012 with 256MB of RAM) made use of I2C bus 0, whereas all other versions of the Pi since have used I2C bus 1. So the above code would change to:

```
sudo i2cdetect -y 0
```

… and you should expect the same output (42) as in the previous step. Additionally you will need to edit line 27 of the Hover_library.py file, changing **bus = smbus.SMBus(1)** to **bus = smbus.SMBus(0)**. A patch that automatically detects the Raspberry Pi version and makes this change for you has been submitted but not yet accepted into the master branch, so this may not be necessary in future versions.

06 Download the sample code

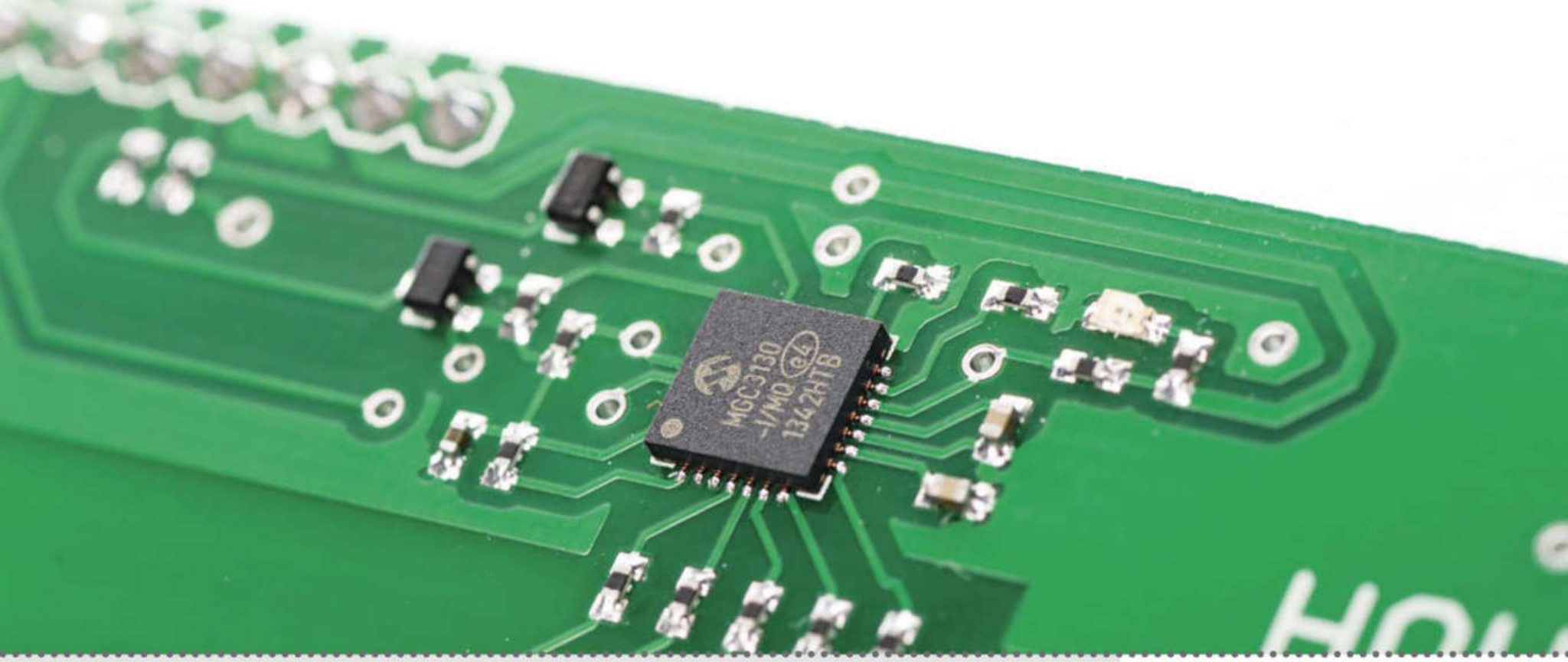
Now you have everything hooked up correctly and your Raspberry Pi is fully up to date, it is time to get the Hover Python library, which makes using the board from Python scripts extremely easy. You can get this using the following command:

```
git clone https://github.com/jonco91/hover_raspberrypi.git
```

This should download a folder called hover_raspberrypi to your /home/pi directory containing all of the files needed for this article. Alternatively you can download the zip file from **bit.ly/1MeuyFQ**.

Why Python?

Python is extremely useful for beginners due to its easy-to-understand syntax, fairly prose-like formation and the flexibility and ease of acquiring existing software libraries to help your projects. It is also the official programming language of the Pi and very well supported. That's not to say that Hover will not work with other programming languages – the creators of Hover just haven't yet released any code libraries in other languages.



Above Five capacitive pads on the board give you five touch events: north, south, east, west and centre

07 Run the example file

The current Hover library is simply a Python file with all of the necessary functions included within it, rather than an installable package (however, this may change in the future). In order to use the functions contained within the Hover_library.py script discussed above, it is therefore necessary to make sure that the Hover_library.py script is located in the same folder as any script you have written that makes use of any of the Hover functions. In a terminal session, navigate to the folder containing the Hover_example.py file and run it using:

```
sudo python Hover_example.py
```

The Hover board will initialise and you will then see a message "Hover is ready", meaning you are good to go.

08 Investigate the output

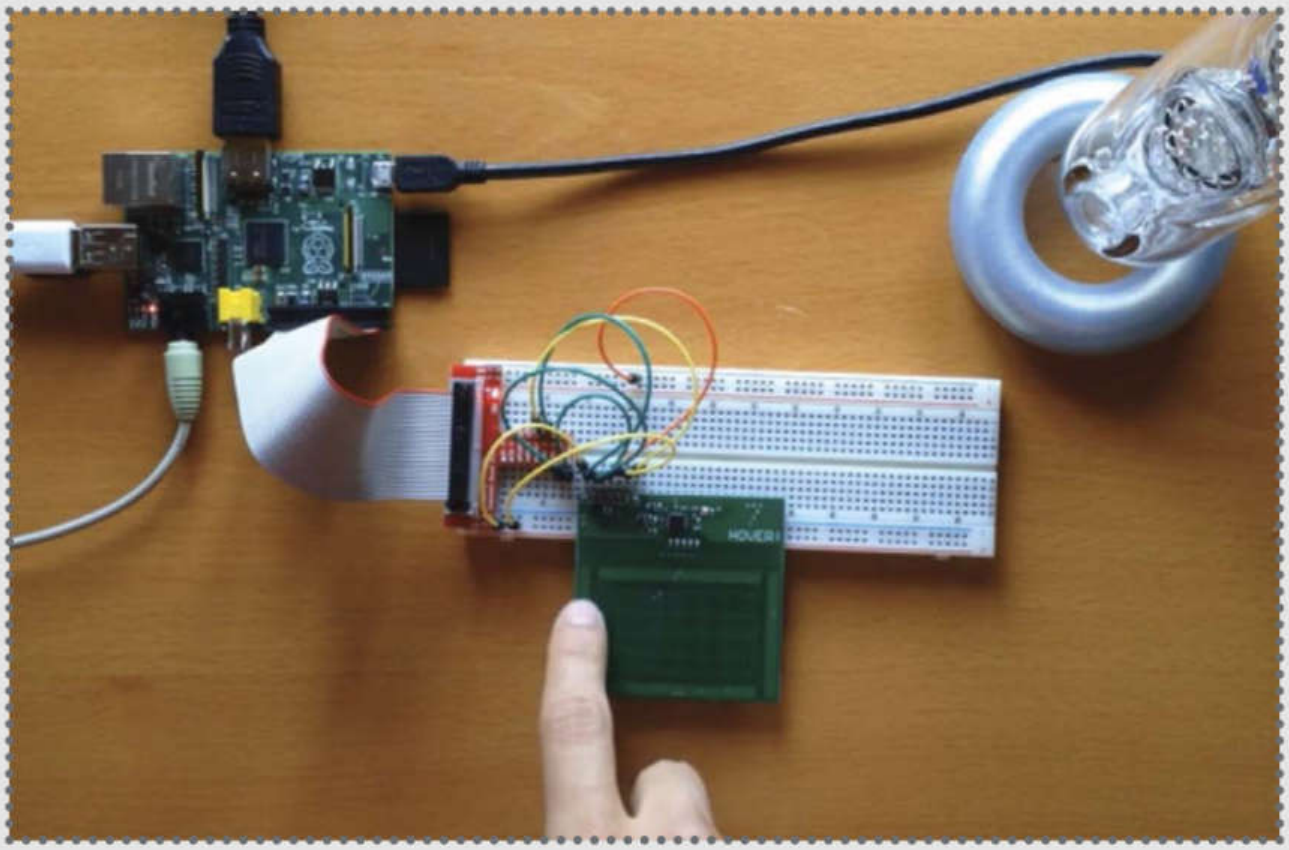
Once you have completed step 7, if you touch the Hover board or make gestures above it you will begin to see output in the terminal, which is a bunch of 0s and 1s, and then a description of what it has seen – right swipe, north tap, etc. The way the Hover works is that it can sense any one of nine different actions and these are

sent to the Raspberry Pi over I2C as an 8-bit binary value. The first three bits describe whether it was a touch or gesture event and the remaining five bits describe the specific type or direction of the event. The breakdown can be seen in the code listing at the end of this tutorial.

09 Enable 3.5mm audio

Grab your speakers and plug them in to the 3.5mm jack plug on the Raspberry Pi. You will then need to route audio to the 3.5mm jack using the following command (you can skip this step if you are using an HDMI display, which has in-built audio):

```
sudo amixer cset numid=3 1
```



10 Make a drum machine

In the hover_raspberrypi folder is another folder called examples that contains code and sounds to turn Hover into a drum machine! Navigate to the hover_raspberrypi directory and then copy the Hover_library.py file into the examples folder by using:

```
cp Hover_library.py examples
```

"It can sense any one of nine different actions and these are sent to the Raspberry Pi over I2C as an 8-bit binary value"

You can then move into the examples folder and run the Hover_drum.py file using:

```
cd examples
sudo python Hover_drum.py
```

Make some gestures and taps on and around Hover and you will have your own basic drum machine!

11 Create your own responses

The great thing about having a Python library available is that it is easy to integrate this device into any of your existing or future projects. The code shown on the next page is all you need to get started with Hover. You will see that on line 15 and onwards there are comments saying "code for … goes here". Essentially, all you need to do is insert the actions you want to occur on the particular event mentioned in the comment and you will be up and running… it really is that easy!

12 Other project ideas

Most of you are probably now wracking your brains for projects you could use Hover in, but let's face it – pretty much any project that requires physical interaction would be made better with touch and gesture control. If you think it is cool but are lacking inspiration, we recommend looking at the projects section of the Hover website at **www.hoverlabs.co/projects**, where there are projects from the Hover creators and community alike. If you make something cool, be sure to send us the pictures!

"Pretty much any project that requires physical interaction would be made better with touch and gesture control"

Where are the hoverboards?

We know you're thinking it and we don't want to leave you disappointed! Hoverboards were first popularised as a mode of personal transportation in *Back To The Future Part II,* looking like a levitating skateboard with no wheels. 25 years later and we're getting close to turning this dream into a reality. Hendo Hoverboards has created a $10,000 hoverboard which uses a principle similar to that of maglev trains to generate lift (**kck.st/ZMd9AA**).

The Code HOVER TEMPLATE

```
import time
from Hover_library import Hover

hover = Hover(address=0x42, ts=23, reset=24)

try:
  while True:

    # Check if hover is ready to send gesture
    # or touch events
    if (hover.getStatus() == 0):
      # Read i2c data and print the type of
      # gesture or touch event
      message = hover.getEvent()
      type(message)
      if (message == "01000010"):
        # code for west touch goes here
      elif (message == "01010000"):
        # code for centre touch goes here
      elif (message == "01001000"):
        # code for east touch goes here
      elif (message == "01000001"):
        # code for south touch goes here
      elif (message == "01000100"):
        # code for north touch goes here
      elif (message == "00100010"):
        # code for swipe right goes here
      elif (message == "00100100"):
        # code for swipe left goes here
      elif (message == "00110000"):
        # code for swipe down goes here
      elif (message == "00101000"):
        # code for swipe up goes here
```

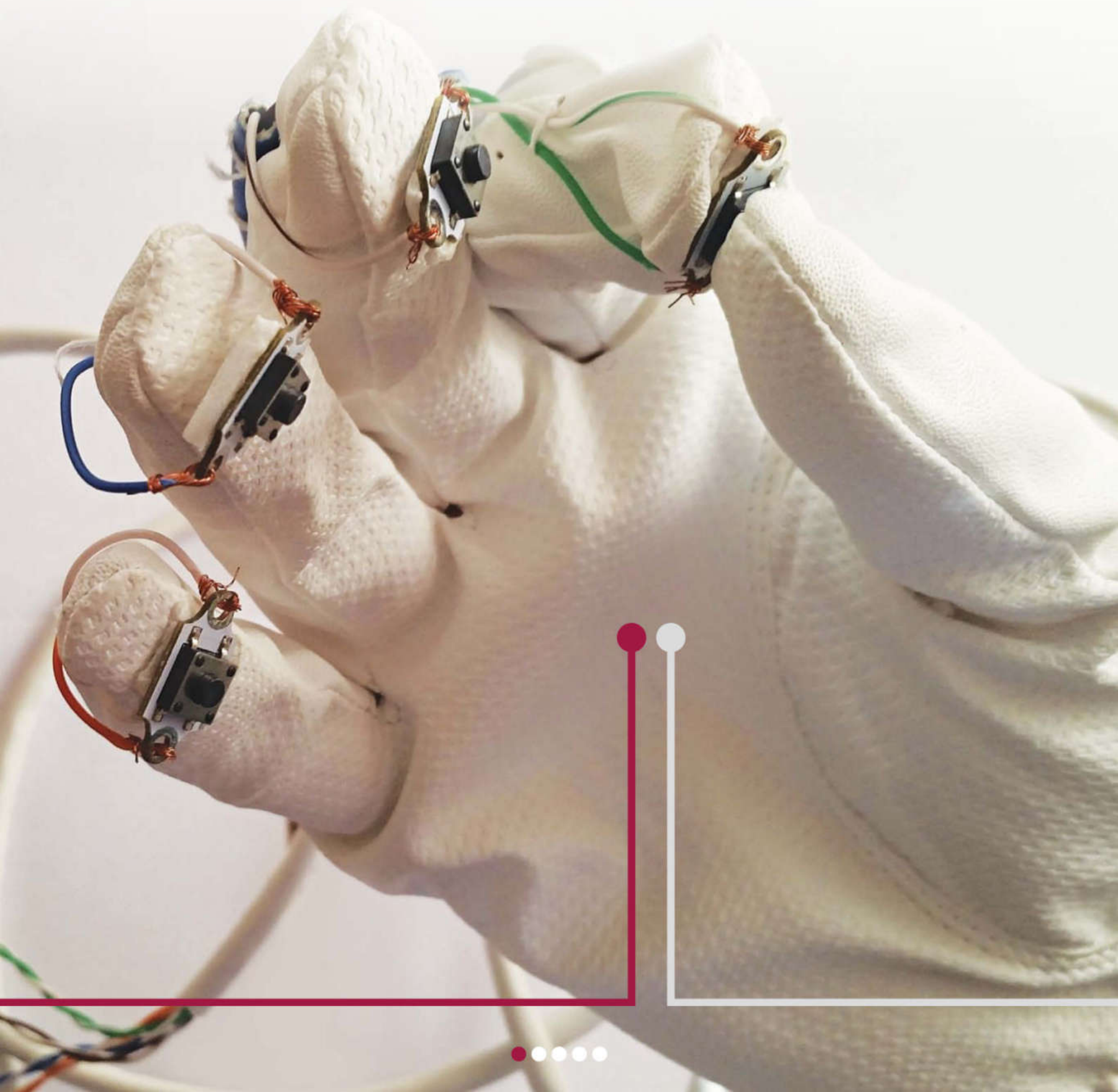
"All you need to do is insert the actions you want to occur on the particular event mentioned in the comment"

The Code HOVER TEMPLATE

```
        # Release the ts pin until Hover is
        # ready to send the next event
        hover.setRelease()
    time.sleep(0.0008)   #sleep for 1ms

except KeyboardInterrupt:
  print "Exiting..."
  hover.end()

except:
  print "Something has gone wrong..."
  hover.end()
```

Pi Glove

Dan Aldred puts the Raspberry Pi camera, Twitter, a music player and a live train timetable under your fingertips



So your Pi Glove grew out of Project New York – tell me about that.

I was invited to the second Picademy back in July 2014. The first day was workshops, education, and the second day was a pure hack day, and we had a bit of a chat with Eben [Upton, from the Raspberry Pi Foundation] and some other people who wanted to support and build something. We wanted to come up with something you could physically use, and we decided to use Scratch GPIO.

So we came up with this idea that if you touched a thumb and finger together, it would move a character on-screen, so that was the original Michael Jackson glove. And at the end we thought that we could actually make something more ergonomic than a phone or a Google Glass – something that fits on the hand – and I said I'd go away and use Python to make something creative for social media, so taking photographs, playing music.

That's how it started. The Project New York element was actually from my son. I showed him the video after Picademy and he asked what I was going to do now, and I said I'd build a glove that lets you do all this stuff. And he said, "Huh, right. We live in a village – this isn't New York." So that was it then – I dedicated the project to New York!

What is the Pi Glove set up to do?

From the index finger, if you press the first button it'll take a picture using picamera. The second button will tweet that picture to your Twitter account (saying something like "This picture was taken at this time"). The third button is an mp3 player; so you've got a list of ten songs on there and it randomly picks one and plays that – that was using Pygame. And then the last one… you know Paul Beech, from Pimoroni? Yeah, so he was on the original Picademy course with me. And after that I did something called Deer

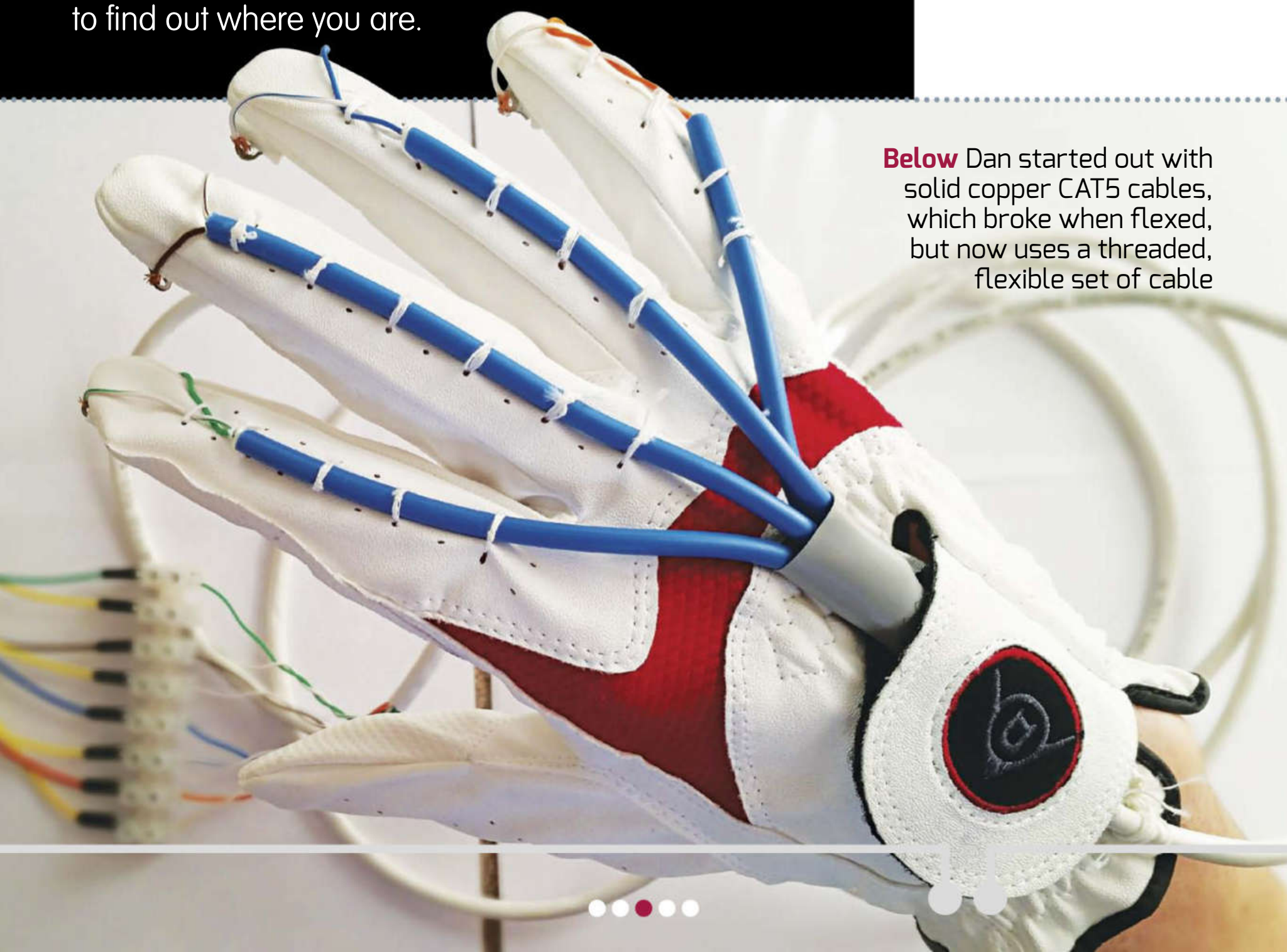
Dan Aldred is a Raspberry Pi Certified Educator and a Lead School teacher for CAS. He is passionate about creating and hacking projects, using projects like this to engage the students he teaches.

Shed, which is kind of a fine technology festival out here in the sticks, and he was there too.

So we were chatting and one of the things we were talking about is train times – I was saying that if you're at the station and want to check when the next train arrives, you have to get your phone out, check the app, load it up – wouldn't it be better if you pressed a button in your pocket and it read it out in your ear, which platform to go to and stuff like that? So he told me to get in touch with this guy who'd done some scraping. Basically, he uses Google Sheets to scrape the data in and that downloads into a text file on the Pi, which then parses through it and reads it out. You've got to be connected to Wi-Fi, and I developed it so that it always works between two stations that you travel between, but the next stage is to develop something to find out where you are.

If you like

Find out more about the Pi Glove on Dan's blog, where he recorded his progress through each stage of development and shared the code that he wrote to solve various challenges:
bit.ly/1zzOwFv



Below Dan started out with solid copper CAT5 cables, which broke when flexed, but now uses a threaded, flexible set of cable

So you have four finger buttons – do you have plans for some kind of hierarchical structure? Maybe a palm button so that if you go down into the mp3 layer, you have volume control, track skipping and so on mapped out?

That will be the next stage. I was chatting to sixth form students about what they want to do, and it's simple things like checking the time. Someone said, "If I wake up in the middle of the night then I usually go for my phone – how good would it be to just press a button and have it tell me the time?" And we got talking about things like if you have special needs or you require assistance, how you could develop for that – you could have some kind of extra button to press that would load up a particular menu.

So I use Python eSpeak to read out what menu you're inside, because there's no GUI or anything to tell you where you are other than this voice saying, "You are now in the camera". It's not a polished article by any means but it works. I was looking online to see what there is already and I was struggling to find stuff – there's an Arduino drum glove which you move up and down, and when you touch your fingers on the desk it plays a drum beat, but there was nothing else like this. Even in terms of the hardware – what is out there? So eventually I envisage having some kind of glove where it's more tactile, rather than pushing buttons.

Have you heard of Imogen Heap and the project MiMu gloves? They're very different, with gesture control as well as touch.

Yes! Well, Sam Aaron, who does Sonic Pi, was at one of the Picademy sessions with Sonic Pi, and on the next day

Further reading

Interested in those MiMu gloves? Check out Imogen Heap performing her song Me, The Machine (**bit.ly/1x2aPUT**), and then visit the MiMu blog if you want to see how those bend sensors actually work: **bit.ly/1sfv8gx**

when he saw the project, he was talking about a friend of his who is trying to get more audience participation in music; so if the audience moves to the left then the volume goes up, and if they move to the right then something else happens. Phenomenal. He was very passionate about the idea of the audience impacting on the music they hear, rather than having 'I'm a performer, I'm just going to play something'. If the audience doesn't like a certain bit and wants to change it, they just walk to the right-hand side and then it alters it.

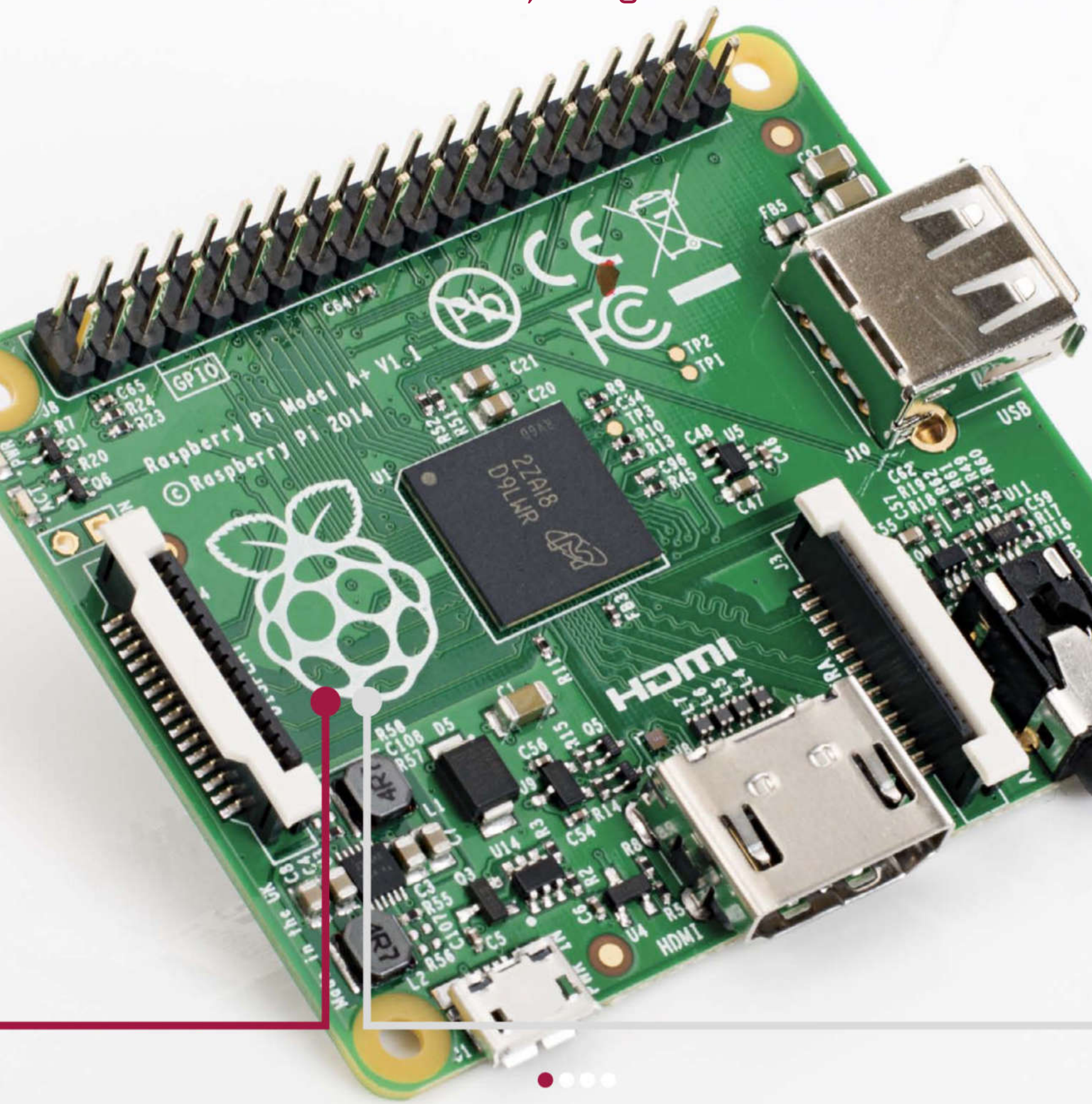
You could add tilt control to the Pi Glove, so it could read if you're tilting to the left or right with a wave.
Well it started with the idea that everyone takes pictures – Instagram is bombarded with pictures – but actually, how many times are people posing for those pictures and how many times are they just clicking away? The original concept was that if I've got to get a phone out to take a picture, okay that's fairly quick but, in twenty to fifty years time, people will look back and say, "Do you remember when we used to have to unlock our phones and put the code in, and we had to wait for this app to load – isn't it better to just press a button and it takes a picture?"

So the idea was that the camera would be located in the breast pocket or something like that – near eye level, taking a picture of what's in front of you – and the issue with that is the quality of the picture, but I think people are becoming so snap-happy that you just want to be able to take photos quickly if something happens. It's ironic, actually – Barack Obama just asked for funding for 50,000 police cameras mounted in their uniforms, to take pictures and stuff like that, so it's the same idea. And I think that's the next step: people walking around with these cameras.

"The idea was that the camera would be located in the breast pocket or something like that – near eye level, taking a picture of what's in front of you"

What is the Model A+?

There's a teeny tiny Raspberry Pi available called the A+, but what's it actually designed to be used for?

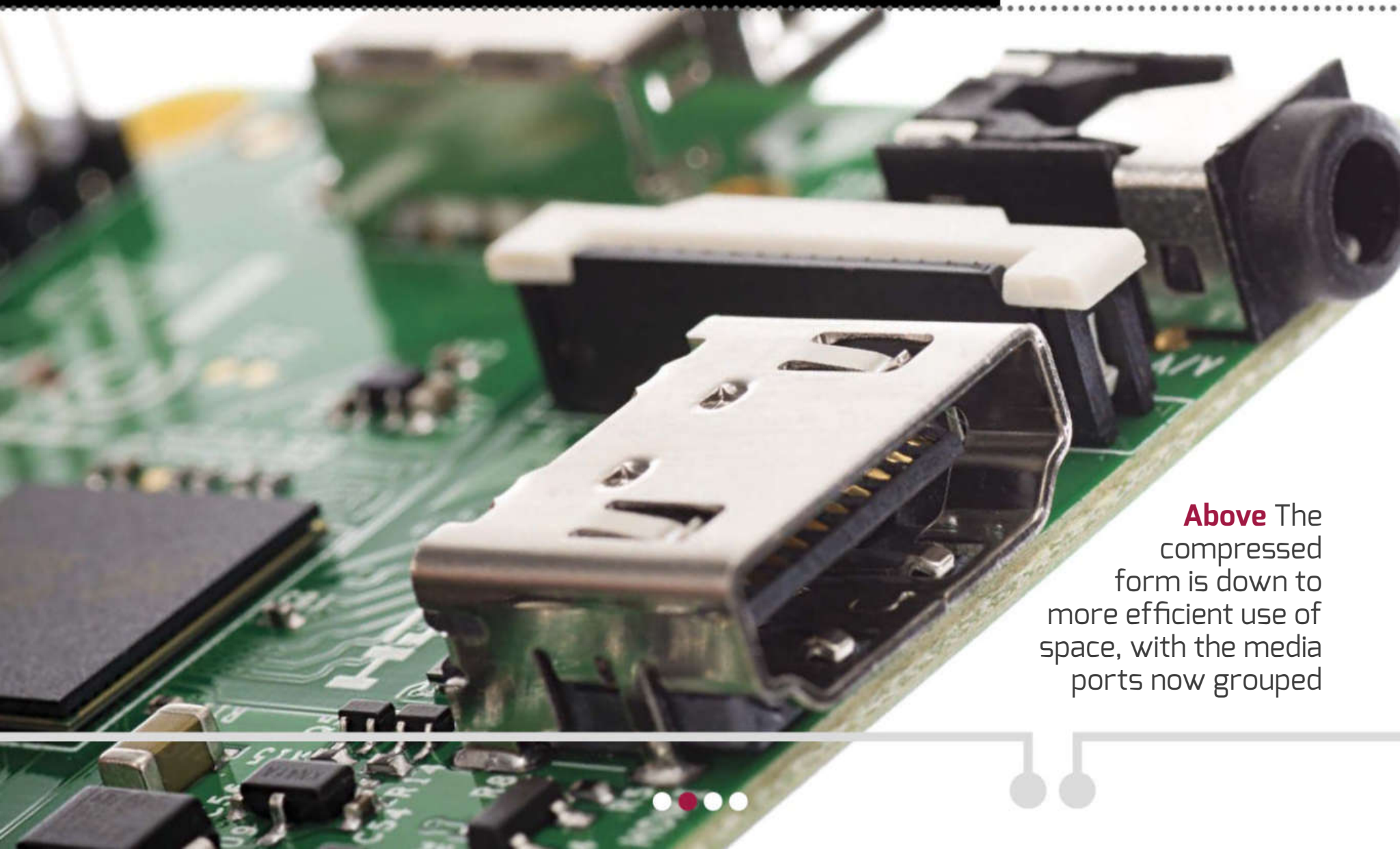


Q Another model of the Raspberry Pi? What's all the fuss with this one then?
A The Raspberry Pi Foundation released a revision of the Raspberry Pi Model A called the Model A+ – basically the same naming convention as the Model B+.

Q I thought the Model B+ was supposed to be the last revision of the original Raspberry Pi?
A It was the last revision of the first Model B board, the flagship and original Raspberry Pi product now replaced by the Raspberry Pi 2 Model B. This is a revision of the Model A board, the lower-spec and cheaper version of the Pi.

Q Is this actually the last revision of the Model A, then?
A The Raspberry Pi folks haven't really said if this is the last version of the Model A. It might well be, but due to the way that people use the Model A it makes sense that there could well be other revisions, with a Raspberry Pi 2 Model A potentially coming out at some point.

"Due to the way that people use the Model A it makes sense that there could well be other revisions, with a Raspberry Pi 2 Model A potentially coming out at some point"



Above The compressed form is down to more efficient use of space, with the media ports now grouped

Q The way that people use the Model A?

A Well, the Model A is a much simpler piece of kit, with fewer inputs and outputs. Specifically, the main difference between it and the Model B was slightly less RAM, no Ethernet port and one less USB port. It still had the same GPIO and processor, though, so it was good for hobby projects as it was cheaper yet could still crunch the same numbers. It just wasn't as good for connecting to a monitor and a keyboard and teaching kids to code.

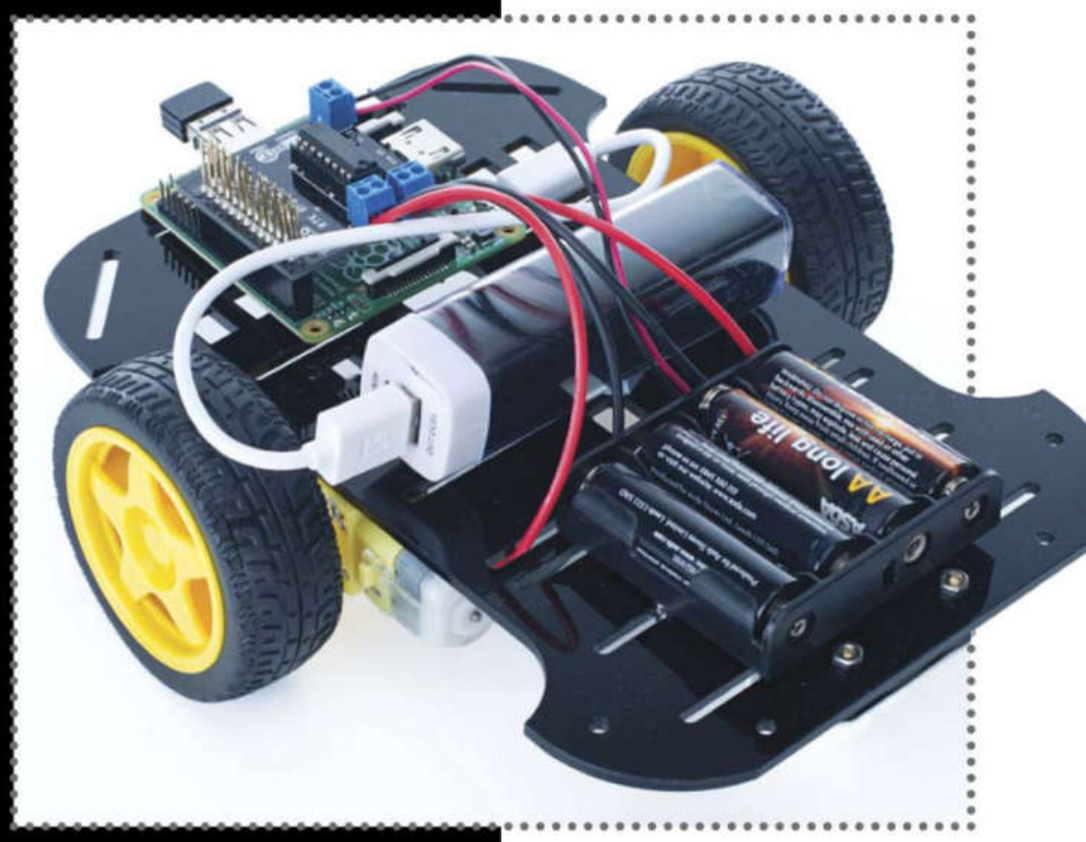
Q Right. So what is the difference between the Model A and Model A+?

A The most noticeable thing about the Model A+ is that it's much smaller than the Model A, which was the same size as the Model B before. It also has the extended GPIO ports of the Model B+, so it now has 40 pins altogether, all of the same layout as the B+. The video-out port has been removed and the audio-out has been moved to sit next to the HDMI port. This 3.5mm jack is also supposed to be of higher quality, so there's less interference on it.

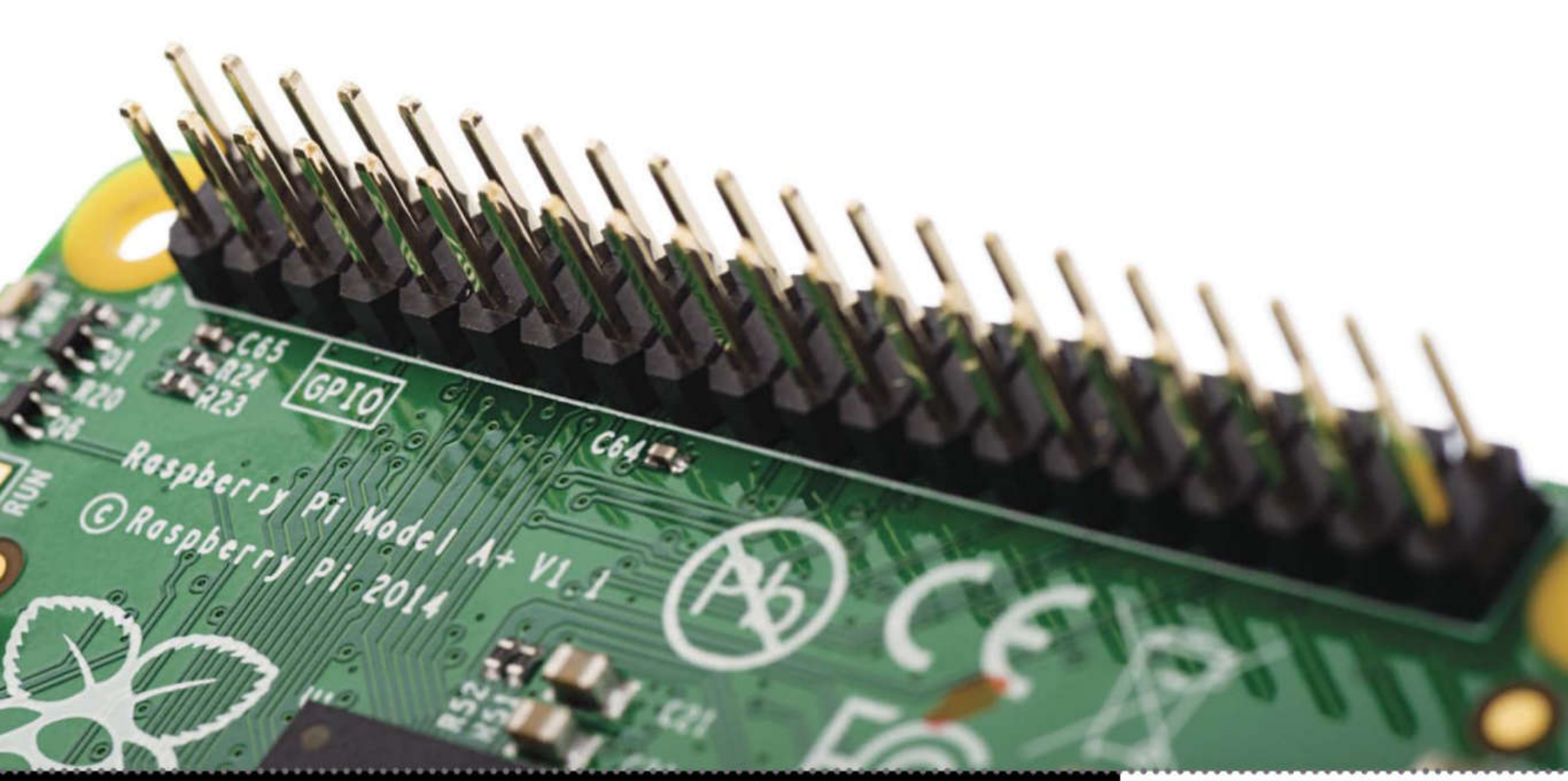
Q How come they were able to make the Model A+ smaller than it was before?

A With the shuffling around of the ports, they were able to shrink it down to the size of the GPIO port. The lack of Ethernet port means that the media ports can occupy the space where it used to be on the normal board, and the USB port fills up the space between the two. It's very neat and compact.

"The main difference between it and the Model B was slightly less RAM, no Ethernet port and one less USB port"



Below Its small size means that the Model A+ is pretty perfect for robotics projects



Q Why didn't they do this with the original Model A?

A It could be a number of reasons: it could be that it was cheaper to produce the same PCB and then add the components afterwards, or it could be that the form factor of both the boards were the same for specific projects and such. For whatever reason, it's been done now and that's pretty cool from our perspective!

Q So the Model A+ will obviously be more expensive to buy, right?

A Actually, no – it's cheaper! It's gone from $25 (£16) to $20 (£13), meaning you have a pretty decent little computer for a lot less. It's much better for certain projects if you just need access to the CPU via the GPIO ports.

Q Is there any special place I can get it?

A The normal places are stocking the A+: element14, RS Components, the Swag Store, etc. Those are the best places to grab them for the moment before they start filtering out to other online and offline stores.

Above The Model A+ now features the full 40 GPIO pins that you get on the B+ and 2B, making it even better for hardware projects

Forecasting the weather with your Raspberry Pi

With Python and a Pi, you can keep an eye on the weather and be prepared for the next big storm

"There are many different services that provide weather data through a number of different APIs"

While people use Raspberry Pis to create lots of applications like Twitter tickers, we thought we'd look at writing up a weather ticker in Python.

There are many services available that provide weather data through a number of different APIs. Some can be accessed directly through a URL, while others are a bit more complicated. Luckily, for the more complicated options there are wrappers written to make the data collection easier. We will look at several services and see how to pull data from them. From there, you can decide how best to display all of this information. The first service we will look at is the one provided by Weather Underground: (**www.wunderground.com**).

This service uses weather information collected by individuals from around the world, using data from personal weather stations. As with most services, you will need to get an API key in order to pull down the weather data for your ticker. You can get a developer key for free, as long as you don't download data more than 500 times per day or 10 times per minute – this should be adequate for personal use. If you need more than this, you can purchase an API key that covers more usage. Interacting with Weather Underground involves sending a request as an HTTP URL and receiving either a JSON or XML file back.

In the sample code, we pull back the data as JSON. The first thing you will need is your location identifier. You can request a one by either latitude and longitude or by geolocating your current IP address. You can even do a search by place name. Once you have this, you can start to make data requests. You can import the urllib2 and json modules to make the request and parse the output. Let's say you wanted to get the current conditions at Cedar Rapids. You could do this with the following Python code:

```
f = urllib2.urlopen('http://api.wunderground.
com/api/YOUR_KEY/geolookup/conditions/q/IA/
Cedar_Rapids.json')
```

This will return a JSON file. Load and parse this data with:

```
json_str = f.read()
json_parsed = json.loads(json_str)
```

The json_parsed variable will now contain all of the available current conditions, such as temperature or precipitation. There are many other data features provided by the Weather Underground, including weather alerts,

"You can get a developer key for free, as long as you don't download data more than 500 times per day or 10 times per minute"

a three-day forecast, a ten-day forecast, hourly forecasts and tide forecasts. There is also historical information, in case you need historical data for some other project.

The next service we will look at is that provided by **https://forecast.io**. Forecast.io aggregates weather data from several different sources around the world. It parses through all of these data sources and provides current weather conditions and a forecast for your location through an API over HTTP. The returned data can be a bit messy, so there are wrappers for many different environments, including Python. This wrapper is available on GitHub from Ze'ev Gilovitz (**https://github.com/ZeevG/python-forecast.io**). While you can download and install from source, you should be able to install it using pip with:

```
pip install python-forecastio
```

As with Weather Underground, you will need to go to the forecast.io site and get an API key in order to make requests. And, as with Weather Underground, this API key is free. Once you import the module, you can call 'load_forecast()' to get the data. The object storing the returned results contains everything available and has several access functions. For example, you can get the hourly forecast with the object function 'hourly()'. These access functions have functions within them to access parts of the sub-data. For example, you can look at the hourly temperature with:

```
byhour = forecast.hourly()
for hourlyData in byhour.data:
   print hourlyData.temperature
```

In most instances, the information available through the wrapper functions should be good enough. But, you

Weather station

To log weather data and send it to Weather Underground, run: **pip install weather**. It contains three submodules that need to be imported individually. The first provides conversion and calculation functions. The second, 'weather.stations', provides the ability to talk to the weather station over a serial connection, but currently only talks to the Vantage and VantagePro. The third, 'weather.services', provides functions to upload your data to the online services.

may have need of more control over your request. In these cases, the forecast.io module has a function called 'manual()'. You can use this to send a particular URL data request to the forecast.io service to pull back the exact information you are interested in.

The last option we will look at is the python-weather-api module. This provides access to weather service providers such as Yahoo! Weather, weather.com and NOAA. This module is actually available as a package within Raspbian. You can install it with the command:

```
sudo apt-get install python-pywapi
```

You can also install it with pip. Once you have it installed, you can import it and request data from any of the three data service providers. The three main functions are:

```
pywapi.get_weather_from_yahoo()
pywapi.get_weather_from_weather_com()
pywapi.get_weather_from_noaa()
```

These essentially get all of the available information from these different servers in a single call. You can then parse the results to pull out the information you are most interested in. The results actually come back as XML data and are then parsed in the return object. You can then pull out the relevant data by using keywords, such as 'forecasts'. Review the module documentation to see what information is available from each of the data sources.

Once you have collected the weather data, you need to display this information. The simplest is to just print it out. This works well if you are using a console as the interface. There are also several LCD options available if you want to make a self-contained weather reporting service. The exact code used to handle the LCD will vary by manufacturer, but they all have very good documentation.

"The last option we will look at is the python-weather-api module. This module provides access to weather service providers such as Yahoo! Weather, weather.com and NOAA"

The Code WEATHER STATION

```
# To talk to Weather Underground we need
# to import modules to handle URLs and
# JSON data
import urllib2
import json

# The next step is to open a URL and
# read the data
f = urllib2.urlopen('http://api.wunderground.com/api/YOUR_KEY/geolookup/
conditions/q/IA/Cedar_Rapids.json')
json_string = f.read()

# Now you can parse the JSON data
# read off the information you need
parsed_json = json.loads(json_string)
location = parsed_json['location']['city']
temp_f = parsed_json['current_observation']['temp_f']

# To talk to forecast.io you need to
# import the forecastio module
import forecastio

# You need your API key and location
apikey = "YOUR_KEY"
latitude = 36.4
longitude = 46.567

# The next step is to load the forecast data
forecast = forecastio.load_forecast(apikey, latitude, longitude)

# You can print out the available hourly data
by_hour = forecast.hourly()
for hourly_data in by_hourly.data:
    print hourly_data
```

The Code WEATHER STATION

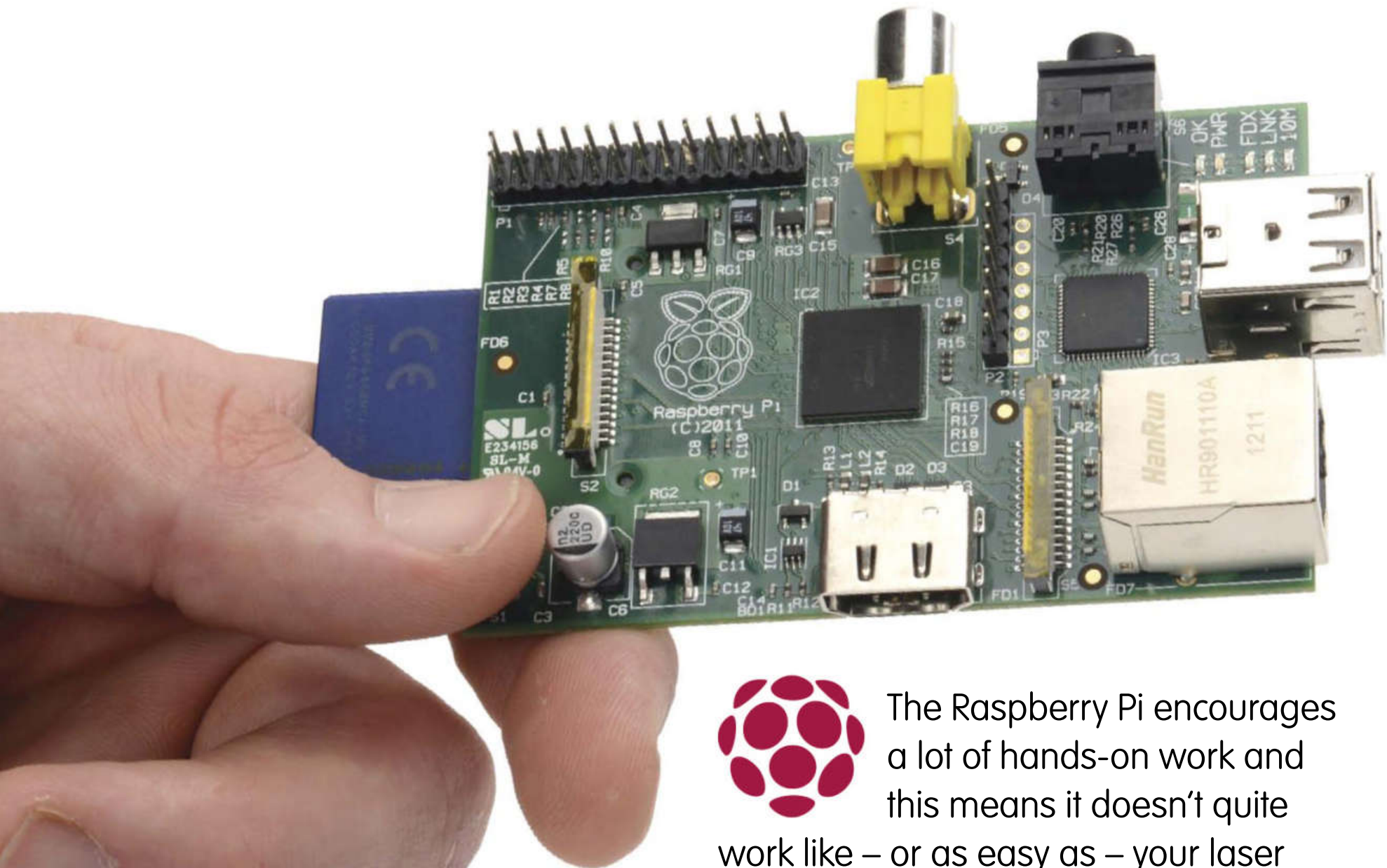
```
# You can also get summaries
by_day = forecast.daily()
print by_day.summary

# To use the Python weather API you need to
# import the pywapi module
import pywapi

# Getting the weather from any of the
# available sources is a single call
# You will need to find and use the
# appropriate location ID

weather_com_result = pywapi.get_weather_from_weather_com('10001')
yahoo_result = pywapi.get_weather_from_yahoo('10001')
noaa_result = pywapi.get_weather_from_noaa('KJFK')

# The data is now in a key/value pair
# structure, ready to read off and used
print weather_com_result['current_conditions']['text']
print yahoo_result['condition']['text']
print noaa_result['weather']
```

"Results come back as XML data and are then parsed in the return object. Pull out the relevant data by using keywords"

Talking Pi

Join the conversation at...

@linuxusermag | Linux User & Developer | @ RasPi@imagine-publishing.co.uk



The Raspberry Pi encourages a lot of hands-on work and this means it doesn't quite work like – or as easy as – your laser focus-tested smartphone interface, or even a normal computer. So we're answering your burning Raspberry Pi questions each issue – get in touch with us on Twitter, Facebook or by email.

What's the best display for my Raspberry Pi?
Rachel via email

Good timing! Last month we would have recommended the PiTFT touchscreen or one of Adafruit's big 10.1-inch displays, depending on what you need to use it for. However, the Raspberry Pi Foundation has just released an official display module for the Pi. It's a 7-inch touchscreen with an 800x480 display, and retails for just $60. It's also super-thin and comes with screws for mounting the Pi and Adapter Board to the back of the screen.

How can I turn things on and off in my kitchen using a Pi?
Simon P via Facebook

Your best bet is Energenie's Pi-mote Infrared Remote Controlled Socket. You plug one of these into your wall socket, then plug your kitchen appliance into the socket on the front of the Pi-mote. Once it's set up with your Pi, you then send infrared signals to the Pi-mote to switch your appliance on or off. The starter kit is £20 and includes two sockets, a controller board and a startup guide (**http://bit.ly/1ihyEmj**).

Keep up with the latest Raspberry Pi news by following @LinuxUserMag on Twitter. Search for the hashtag #RasPiMag

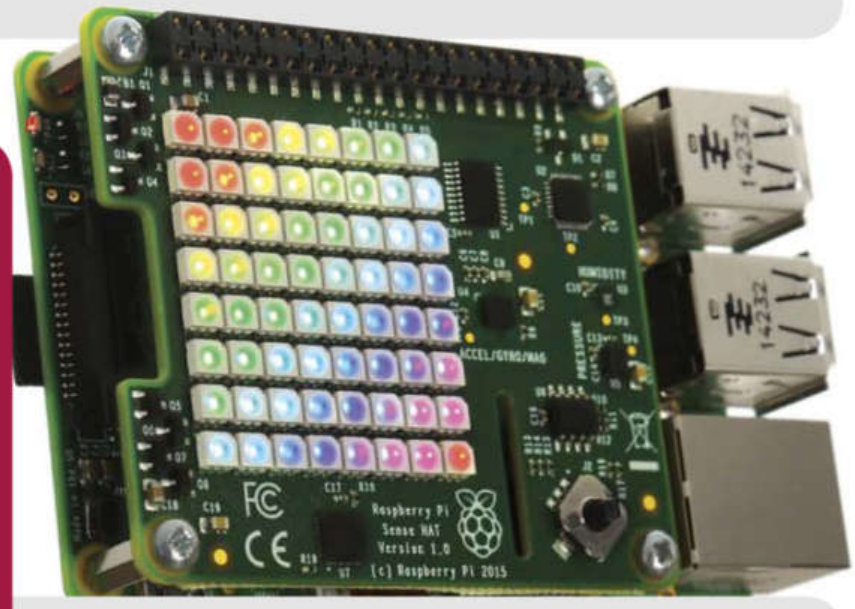
JUST A SCORE
WHAT'S YOUR JUST A SCORE?

Have you heard of Just A Score? It's a new, completely free app that gives you all the latest review scores. You can score anything in the world, like and share scores, follow scorers for your favourite topics and much more. And it's really good fun!

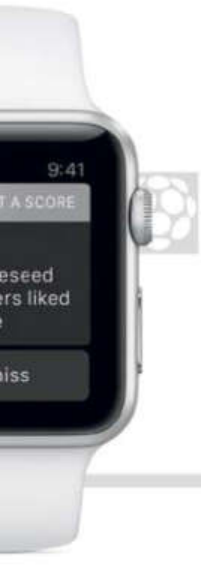
I want to make games. Where should I start?
Tim via Twitter

Check out the Pygame Zero library: **http://bit.ly/1Ohn87H**. It makes it far easier for you to create games using the popular Pygame modules – and even better, it's intended to be used as an educational tool, so work your way through the docs. Later, once you've got the hang of it, you can move on from Pygame Zero and start learning how to do things like event loops manually with just Pygame.

I got one of those Sense HATs. What's a cool project to try out?
Stacey via Facebook



Hopefully you'll enjoy the ISS tracker project in this issue! The Foundation also has a great resources page that's full of Sense HAT guides and cheat sheets (**https://bit.ly/1Qw2yy8**). If you're feeling adventurous, you could even take a look at the National STEM Centre's resources site (**https://bit.ly/1Qw2K0k**). Once you're up to speed, you could then check out the other competition entries and have a go at working those out (**https://bit.ly/1IYw0MJ**).



JUST A SCORE
WHAT'S YOUR JUST A SCORE?

You can score absolutely anything on Just A Score. We love to keep an eye on free/libre software to see what you think is worth downloading...

10 LinuxUserMag scored **10** for Keybase

9 LinuxUserMag scored **9** for Cinnamon Desktop

8 LinuxUserMag scored **8** for Tomahawk

4 LinuxUserMag scored **4** for Anaconda installer

3 LinuxUserMag scored **3** for FOSS That Hasn't Been Maintained In Years

SCORE ANYTHING
JUST A SCORE
Download on the App Store

Next issue

Get inspired **Expert advice** **Easy-to-follow guides**

SWITCH YOUR PC FOR A

RASPBERRY PI 2

Get this issue's source code at:

www.linuxuser.co.uk/raspicode